

Université de Sherbrooke

**Ajout de la diffusion dans la modélisation  
pharmacocinétique du rehaussement pour l'imagerie par  
résonance magnétique dynamique.**

par

**Martin Pellerin**

Département de médecine nucléaire et radiobiologie

Mémoire présenté à la faculté de médecine  
et des sciences de la santé  
en vue de l'obtention du grade de  
maître ès sciences (M.Sc.) en radiobiologie

11 mai 2007



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file    Votre référence*

*ISBN: 978-0-494-31432-6*

*Our file    Notre référence*

*ISBN: 978-0-494-31432-6*

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

## TABLE DES MATIÈRES

---

Table des matières .....	II
Liste des illustrations .....	III
Liste des sigles, abréviations et symboles .....	IV
Résumé.....	1
Introduction.....	4
Théorie.....	7
Imagerie par résonance magnétique (IRM) .....	7
Images pondérées en $T_1$ (État stationnaire) .....	8
Détermination de $T_1$ .....	11
Agents de contraste .....	12
Imagerie par résonance magnétique dynamique rehaussée par agent de contraste (IRM-dynamique).....	14
Modélisation pharmacocinétique.....	16
Modèle proposé .....	22
Algorithmes génétiques.....	28
Recuit simulé .....	30
Matériels et méthodes .....	33
Simulation.....	33
Outils informatiques : .....	36
Optimisation par algorithmes génétiques .....	37
Optimisation par recuit simulé.....	40
Modèle animal .....	42
IRM-dynamique.....	43
Résultats et discussions .....	46
Algorithme génétique .....	46
Recuit simulé .....	51
Publication : Incorporating Contrast Agent Diffusion into the Analysis of DCE-MRI Data .....	53
Conclusions .....	89
Remerciements.....	92
Liste des références.....	93
Annexe A : Images pondérées par la diffusion .....	A1
Annexe B : Définitions de matrice.....	B1
Annexe C : Code pour l'algorithme génétique.....	C1
Annexe D : Code pour le recuit simulé .....	D1

## LISTE DES ILLUSTRATIONS

---

Figure 1 .....	9
Figure 2 .....	12
Figure 3 .....	16
Figure 4 .....	21
Figure 5 .....	23
Figure 6 .....	33
Figure 7 .....	34
Figure 8 .....	35
Figure 9 .....	47
Figure 10 .....	48
Figure 11.....	50
Figure 1 (Publication) .....	65
Figure 2 (Publication).....	70
Figure 3 (Publication).....	71
Figure 4 (Publication).....	74
Figure 5 (Publication).....	75
Figure 6 (Publication).....	77
Figure 7 (Publication).....	78
Figure A1 .....	A3

## **LISTE DES SIGLES, ABRÉVIATIONS ET SYMBOLES**

---

$a$	Largeur d'un pixel carré [mm].
AC	Agent de contraste.
AG	Algorithme génétique.
AIF	Fonction d'entrée artérielle (Arterial input function) [mM].
$b$	Facteur $b$ caractérisant les conditions d'une séquence pondérée par la diffusion de type Stejskal-Tanner [s mm <sup>-2</sup> ].
$B_0$	Champ magnétique primaire d'un appareil de résonance magnétique [T].
$C(t)$	Concentration d'agent de contraste dans le tissu [mM].
CP	Probabilité de <i>crossover</i> utilisée dans les algorithmes génétiques (Crossover probability).
$C_p(t)$	Concentration d'agent de contraste dans le plasma sanguin [mM].
$D$	Coefficient de diffusion de l'agent de contraste [mm <sup>2</sup> s <sup>-1</sup> ].
$\bar{D}$	Matrice creuse décrivant le processus de diffusion.
$e$	Épaisseur de la tranche associée à une image [mm].
$E$	Fraction d'extraction.

EEE	Espace extravasculaire extracellulaire.
$F(S)$	Fonction à minimiser pour trouver la meilleure solution (appelée aptitude d'une solution).
FOV	Champ de vue (Field of view).
Fsems	Écho de spin accéléré multi-tranches (Fast Spin Echo Multi-Slice).
$G_{Diff}$	Intensité des gradients d'encodage d'une séquence pondérée par la diffusion de l'eau [gauss cm <sup>-1</sup> ].
Gems	Écho de gradient multi-tranches (Gradient Echo Multi-Slice).
$\bar{K}$	Matrice diagonale contenant les valeurs de $K^{trans}$ .
Hct	Hématocrite.
IRM	Imagerie par résonance magnétique.
$K^{trans}$	Taux de transfert transcapillaire [min <sup>-1</sup> ].
$m$	Nombre d'encodages de fréquence d'une image d'IRM.
$M_0$	Magnétisation à l'équilibre.
$M_z$ et $M_{\perp}$	Magnétisation longitudinale et transversale respectivement.
$n$	Nombre d'encodages de phase d'une image d'IRM.
NA	Nombre d'acquisition.
PS	Produit de la perméabilité des capillaires sanguins et de leur surface par unité de masse [cm <sup>3</sup> min <sup>-1</sup> g <sup>-1</sup> ].

PS	Taille de la population utilisée dans les algorithmes génétiques (Population Size).
$r$	Relaxivité de l'agent de contraste [ $s^{-1} \text{ mM}^{-1}$ ].
$R_1$	Taux de relaxation longitudinale ( $1 / T_1$ ) [ $s^{-1}$ ].
RF	Radio fréquence.
RR	Région de référence.
RS	Recuit simulé.
$S$	Signal de sortie de l'appareil de résonance magnétique [u.a.].
$S_0$	Paramètre relié au signal potentiel d'un voxel [u.a.].
$S$	Solution.
$\bar{S}$	Surface orientée entre deux voxels voisins.
SemsDW	Écho de spin pondéré par la diffusion (Diffusion Weighted Spin Echo Multi-Slice).
$T$	Paramètre de contrôle de l'algorithme de recuit simulé.
$T_1$	Temps de relaxation longitudinale [s].
$T_2$ et $T_2^*$	Temps de relaxation transversale [s].
$T_E$	Temps d'écho [ms].
$T_R$	Temps de répétition [ms].
$V$	Volume d'un voxel.

$\bar{\mathbf{V}}$	Matrice diagonale contenant l'inverse des valeurs de $v_e$ .
$v_e$	Fraction de volume extravasculaire extracellulaire [sans unité].
$\alpha$	Angle d'excitation de la magnétisation [ $^\circ$ ].
$\delta$	Durée des gradients d'encodage d'une séquence pondérée par la diffusion de l'eau [ms].
$\Delta$	Délais entre les gradients d'encodage d'une séquence pondérée par la diffusion de l'eau [ms].
$\varepsilon$	Paramètre régulant la descente du paramètre de contrôle de l'algorithme de recuit simulé [sans unité].



## **Ajout de la diffusion dans la modélisation pharmacocinétique du rehaussement pour l'imagerie par résonance magnétique dynamique.**

L'imagerie par résonance magnétique dynamique (IRM-dynamique) consiste en l'observation de la distribution dans le temps d'un agent de contraste à l'aide de l'IRM. L'une des approches très répandues est d'analyser les données à l'aide de modèles mathématiques qui décrivent la pharmacocinétique de cet agent dans les tissus. L'une des hypothèses utilisées par l'ensemble des modèles présentés dans la littérature à ce jour est que les images d'IRM-dynamique peuvent être analysées pixel-par-pixel ce qui néglige implicitement la possibilité de diffusion de l'agent à l'intérieur des tissus.

Dans ce mémoire, un nouveau modèle est proposé dans lequel la diffusion de l'agent est explicitement incluse dans un modèle à deux compartiments. Les deux paramètres couramment utilisés dans la littérature sont :  $K^{\text{trans}}$ , le taux de transfert transcapillaire, et  $v_e$ , la fraction de volume extravasculaire extracellulaire. Deux méthodes d'optimisation stochastique ont été évaluées pour le lissage avec le modèle proposé à cause de la très grande taille de l'espace des solutions. Le modèle a été caractérisé avec des données simulées incluant la diffusion de l'agent de contraste et des données

expérimentales montrant des signes de diffusion à l'intérieur du tissu tumoral.

Les résultats avec les données simulées montrent que le modèle peut retrouver de façon fiable les valeurs des paramètres utilisés pour générer ces données (erreur relative moyenne de 16% pour  $K^{\text{trans}}$  et 17% pour  $v_e$ ) même lorsqu'un niveau de bruit raisonnable est ajouté. Le modèle standard à deux compartiments négligeant la diffusion retourne des distributions de valeurs de  $K^{\text{trans}}$  erronées (erreur relative moyenne de 43%) qui sont moyennée sur le tissu.

Lorsque les données expérimentales sont ajustées avec le modèle standard, les valeurs de  $K^{\text{trans}}$  retournées montrent une perfusion moyennée sur l'ensemble du tissu, ce qui n'est pas en accord avec le rehaussement initial du signal qui est observé. À l'opposé, le modèle proposé retourne des cartes de  $K^{\text{trans}}$  ayant une démarcation franche entre les zones bien perfusées et celles très peu perfusées en accord avec ce qui est observé sur les images d'IRM. De plus, le modèle standard à deux compartiments assigne des valeurs n'ayant pas de sens physique à  $v_e$  ( $v_e > 1$ ) dans le centre des tumeurs où l'agent parvient par diffusion à partir de la périphérie bien vascularisée. De son côté, le modèle proposé trouve des valeurs plausibles de  $v_e$  pour l'ensemble du tissu.

**Mots clés :** IRM, diffusion, perfusion, modélisation pharmacocinétique, DCE-MRI

## INTRODUCTION

---

L'imagerie par résonance magnétique (IRM) a été inventée par Paul Lauterbur, Peter Mansfield et Raymond Damadian dans les années 70 (LAUTERBUR, 1973, GARROWAY *et al.*, 1974, DAMADIAN, 1971). Pour leurs travaux, qu'ils ont effectués séparément, les deux premiers ont reçu le prix Nobel de médecine en 2003. L'omission de R. Damadian lorsque le prix a été remis a engendré une importante controverse. Depuis son avènement, l'IRM a connu beaucoup de développements et les nouvelles applications qui ont révolutionné la médecine moderne.

Aujourd'hui, l'IRM est utilisée pour le dépistage et la morphologie de tumeurs non seulement au cerveau où l'IRM est la seule modalité d'imagerie qui offre du contraste entre la matière blanche et la matière grise, mais aussi dans le reste du corps comme dans le sein ou la prostate. La très grande variété de sources de contraste permet à l'IRM de baser son analyse sur bien plus qu'une seule caractéristique des tissus. En effet, l'imagerie fonctionnelle cartographie l'activité au cerveau en observant les changements de signal engendrés par la consommation de l'oxygène (HAACKKE *et al.*, 1999). D'un autre côté, l'IRM-dynamique utilise l'administration d'agents de contraste (AC) pour décrire l'état de la vascularisation de lésions au cerveau

engendrées par la sclérose en plaque ou encore de tumeurs dans l'ensemble du corps (TOFTS *et al.*, 1999).

Cette dernière application de l'IRM nécessite l'acquisition de plusieurs images successives pour suivre l'entrée et l'élution de l'AC dans les tissus. L'utilisation de modèles pharmacocinétiques mathématiques permet de caractériser la micro-vascularisation dans les lésions. Ces modèles à compartiments décrivent les échanges entre le milieu sanguin et les tissus à l'aide de paramètres associés à des taux d'échange, des fractions de volume du tissu ou des temps de vie à l'intérieur des compartiments (TOFTS *et al.*, 1999, YANKEELOV *et al.*, 2003). L'un des phénomènes qui n'a pas encore été abordé par les modèles associés à l'IRM-dynamique est la possibilité que l'AC diffuse à l'intérieur d'un tissu pour passer d'une région bien perfusée vers une autre où la vascularisation est déficiente ou absente. Pourtant ce phénomène a été documenté dans la littérature (JAIN, 1987) et peut être observé dans les images de tumeurs (SU *et al.*, 2003) (CHA *et al.*, 2003). Dans ce mémoire, une approche pour inclure explicitement la diffusion de l'AC dans le modèle standard de Tofts (TOFTS *et al.*, 1999) est proposée.

Dans un premier temps, la théorie associée au contraste présent dans les images pondérées en  $T_1$  et à l'utilisation d'AC est abordée. L'essentiel des différents modèles utilisés dans la littérature depuis les débuts de l'IRM-

dynamique est résumé. Les détails permettant d'inclure la diffusion dans un modèle pharmacocinétique sont aussi abordés.

L'addition de la diffusion dans un modèle crée une interaction entre les pixels d'une image qui amplifie la complexité de l'ajustement des données. Des approches stochastiques d'optimisation qui diffèrent des techniques standards sont abordées pour tenter de résoudre le problème de l'ajustement des données. En particulier, les algorithmes génétiques (AG) et le recuit simulé (RS) sont traités.

Le comportement du modèle couplé aux AG est évalué à l'aide de données simulées. L'utilisation du RS sera évaluée à l'aide de données simulées et de données tirées d'images d'adénocarcinomes mammaires de souris. L'ensemble des résultats sur le RS est exposé dans une publication reportée en totalité dans la section Résultats de ce mémoire.

## THÉORIE

---

### **Imagerie par résonance magnétique (IRM)**

L'imagerie par résonance magnétique repose sur la manipulation de la magnétisation des noyaux d'hydrogène présents dans les molécules d'eau. En présence d'un champ magnétique ( $B_0$ ) un déséquilibre entre les deux états quantiques possibles du spin des noyaux d'hydrogène  $\{+1/2, -1/2\}$  crée une magnétisation dans les tissus biologiques qui peut être exploitée pour créer des images d'une grande précision. Cette magnétisation et le signal associé sont d'autant plus importants que le champ  $B_0$  est intense.

Contrairement à la tomographie d'émission de positron qui nécessite l'utilisation de radio-traceurs, l'IRM n'a pas besoin de l'injection d'une molécule pour produire des images. La majorité des formes de contraste exploitées en IRM sont intrinsèques aux tissus mesurés. Malgré tout, certains agents de contraste peuvent être ajoutés aux tissus pour en changer les propriétés magnétiques et ainsi affecter leur signal.

La description complète du fonctionnement de l'IRM dépasse le cadre de cette section théorique. Pour plus d'information se référer à HAACKE *et al.*, 1999 ou NISHIMURA, 1996.

## Images pondérées en $T_1$ (État stationnaire)

Il existe plusieurs contrastes en IRM :  $T_1$ ,  $T_2$ , la densité de proton et la diffusion apparente de l'eau pour ne nommer que ceux-là. Les deux premiers sont les plus couramment utilisés et sont associés directement au comportement de la magnétisation de l'eau dû aux interactions avec le milieu. Leur origine est bien décrite de façon phénoménologique par les équations de Bloch. Ces équations décrivent la relaxation longitudinale (selon l'axe du champ  $B_0$ ) et transversale (perpendiculairement à l'axe de  $B_0$ ) :

$$\frac{dM_z}{dt} = \frac{1}{T_1}(M_0 - M_z) \quad [1]$$

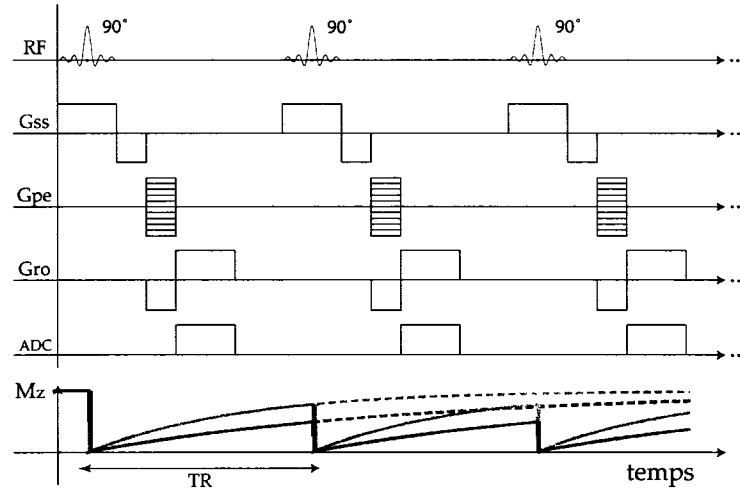
$$\frac{dM_{\perp}}{dt} = -\frac{1}{T_2} M_{\perp} \quad [2]$$

où  $M_z$  est la composante de la magnétisation qui est colinéaire avec le champ  $B_0$ ,  $M_0$  est l'amplitude maximale de la magnétisation à l'équilibre thermodynamique et  $M_{\perp}$  est la magnétisation perpendiculaire au champ  $B_0$ .  $T_1$  et  $T_2$  sont les temps de relaxation longitudinale et transversale, respectivement.

De la première équation, on peut tirer le comportement du signal de résonance magnétique nucléaire d'un petit volume, un voxel, pour une séquence de pulses telle qu'illustrée à la figure 1. L'utilisation d'un temps de



répétition,  $T_R$ , court par rapport à la valeur de  $T_1$  des tissus mesurés fait que les excitations radiofréquence (RF) sont répétées plus rapidement que le temps nécessaire à la magnétisation pour retourner à l'équilibre. Le contraste provient du rapport entre le temps de répétition et  $T_1$ .



**Figure 1 :** Séquence d'un écho de gradient utilisée pour générer des images pondérées en  $T_1$ . Le temps de répétition court ( $T_R$ ) ne permet pas à la magnétisation longitudinale ( $M_z$ ) de retourner à son amplitude maximale entre deux excitations RF.

En effet, la solution à l'équation différentielle décrivant le comportement de la magnétisation longitudinale (Eq. [1]) est :

$$M_z(t) = M_0(1 - e^{-t/T_1}) + M_z(0)e^{-t/T_1} \quad [3]$$

Ainsi dans le cas d'une séquence d'échos de gradient avec  $T_R \ll T_1$ , la condition suivante régie le comportement de la magnétisation juste avant le nième pulse RF d'excitation :

$$M_z(nT_R^-) = M_0(1 - e^{-T_R/T_1}) + M_z((n-1)T_R^-)e^{-T_R/T_1} \cos(\alpha) \quad [4]$$

où  $\alpha$  est l'angle auquel la magnétisation est basculée par l'excitation RF. La notation  $T_R^-$  identifie le temps tout juste avant l'excitation RF. Puisque l'état stationnaire est atteint rapidement, *i.e.*  $M_z(nT_R^-) = M_z((n-1)T_R^-)$ , la condition devient :

$$M_z(nT_R^-)(1 - e^{-T_R/T_1} \cos(\alpha)) = M_0(1 - e^{-T_R/T_1}) \quad [5]$$

$$M_z(nT_R^-) = \frac{M_0(1 - e^{-T_R/T_1})}{1 - e^{-T_R/T_1} \cos(\alpha)} \quad [6]$$

L'amplitude du signal détecté,  $S$ , durant l'expérience, qui dépend de la magnétisation totale à l'équilibre, du gain de l'appareil et de la configuration spatiale de l'échantillon, est donné par :

$$|S(nT_R + t)| \propto M_z(nT_R^-) \sin(\alpha) e^{-t/T_2} \quad [7]$$

$$|S(nT_R + t)| = S_0 \frac{1 - e^{-T_R/T_1}}{1 - e^{-T_R/T_1} \cos(\alpha)} \sin(\alpha) e^{-t/T_2} \quad [8]$$

où  $S_0$  inclut la densité de protons, le gain de l'appareil et la configuration spatiale de l'échantillon. La décroissance du signal due aux inhomogénéités du champ magnétique, caractérisée par le temps de relaxation  $T_2^*$ , a été omise pour simplifier l'explication.

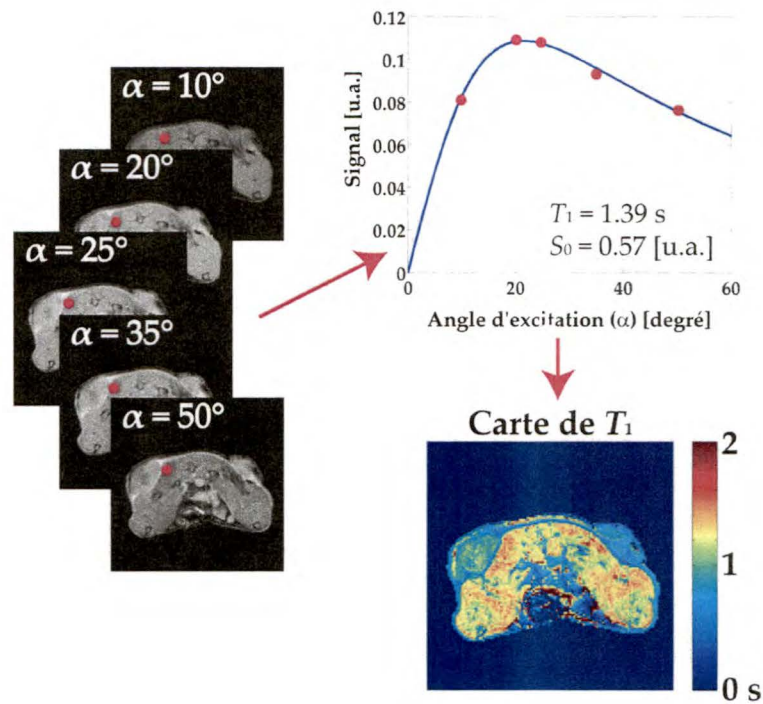
En prenant, le temps d'écho,  $T_E$ , beaucoup plus petit que la valeur de  $T_2$  des tissus mesurés, la décroissance exponentielle du signal après le pulse RF d'excitation perd son influence et le signal d'un voxel devient uniquement dépendant de  $T_R$ ,  $T_1$  et  $\alpha$  tel que présenté dans l'Eq. [9].

$$S = S_0 \frac{1 - e^{-T_R/T_1}}{1 - e^{-T_R/T_1} \cos(\alpha)} \sin(\alpha) \quad [9]$$

Dans des images pondérées  $T_1$ , le contraste provient des variations de  $T_1$  pour différents tissus. En effet,  $S_0$  est un paramètre assez uniforme pour beaucoup de tissus mous. Ainsi des tissus avec une valeur de  $T_1$  courte auront un signal plus intense que les tissus avec un  $T_1$  long (Pour plus d'information se référer à HAACKE *et al.*, 1999).

## Détermination de $T_1$

Il est possible de déterminer la valeur de  $T_1$  pour chaque voxel en prenant plusieurs images pondérées en  $T_1$  avec différents angles d'excitation et en gardant tous les autres paramètres constants. En effet l'ajustement de ces données avec l'Eq. [9] pour chacun des voxels retournera une carte des valeurs de  $T_1$  (Fig.2 ).



**Figure 2 :** L'utilisation de plusieurs angles d'excitation pour générer des images pondérées en  $T_1$  permet de retrouver la valeur de  $T_1$  pour chaque voxel en ajustant les données avec l'Eq. [9].

## Agents de contraste

Bien que l'utilisation d'AC en IRM ne soit pas nécessaire pour l'obtention d'images utiles au diagnostic, certaines pathologies peuvent être identifiées plus facilement en utilisant des molécules ayant une influence sur la magnétisation de l'eau. Ainsi certains agents peuvent diminuer les valeurs de  $T_1$  (Magnevist<sup>TM</sup> (Berlex), Gadovist<sup>TM</sup> (Schering), Vasovist<sup>TM</sup> (Epix)) ou de  $T_2$  (Feridex<sup>TM</sup> (Berlex)) des molécules d'eau en contact avec ceux-ci. Les

molécules contenant des atomes fortement paramagnétiques comme le Gd-DTPA (gadolinium-diethylene-triamine penta-acetic acid, Magnevist™ (Berlex)) diminuent  $T_1$  dans les tissus où elles se logent. La valeur de  $T_1$  est déterminée par l'équation suivante (TOFTS, 1997) :

$$\frac{1}{T_1(t)} = R_1(t) = R_{1,0} + r_1 C(t) \quad [10]$$

où  $r_1$  est la relaxivité de l'agent ( $r_1 = 4.5 \text{ [s}^{-1} \text{ mM}^{-1}]$  à 1.5 Tesla pour le Magnevist™ dans une solution saline (STANISZ et HENKELMAN, 2000), et  $R_{1,0}$  est la valeur du taux de relaxation longitudinale avant l'injection de l'AC. L'Eq. [9] stipule que les tissus avec un  $T_1$  court auront un signal plus intense pour un  $S_0$  équivalent. Ainsi le Gd-DTPA a pour effet de rehausser le signal des tissus où il se loge dans des images pondérées en  $T_1$ .

L'administration du Gd-DTPA se fait par voie intraveineuse, il se distribue dans l'organisme grâce à la circulation sanguine et passe dans l'espace extravasculaire extracellulaire (EEE) au niveau des capillaires sanguins. La molécule de Gd-DTPA ne pénètre pas dans le milieu cellulaire (TOFTS, 1997) et son comportement dans l'organisme n'est dirigé par aucune interaction biochimique. Dès le premier passage dans les reins, le Gd-DTPA commence à être éliminé de l'organisme et à s'accumuler dans la vessie. Avec le temps, les molécules logées dans l'EEE retournent dans la circulation

sanguine et sont, elles aussi, éliminées par les reins (élimination de  $83\% \pm 14\%$  en 6 heures et  $91\% \pm 13\%$  en 24 heures. Tiré de la fiche technique du Magnevist™).

Cet AC est utilisé en clinique pour la détection de tumeurs. En effet, la vascularisation déficiente et très perméable qui caractérise typiquement le microenvironnement tumoral (JAIN, 2001) ainsi qu'une fraction de volume extravasculaire extracellulaire élevée (DONAHUE *et al.*, 1995) font que les molécules de Gd-DTPA s'accumulent dans ces tissus et diminuent de beaucoup leur valeur de  $T_1$ . Les tumeurs paraissent alors « blanches » sur des images pondérées en  $T_1$ .

### **Imagerie par résonance magnétique dynamique rehaussée par agent de contraste (IRM-dynamique)**

L'IRM-dynamique est utilisée pour observer la façon dont le rehaussement des tissus se fait après l'injection d'un agent de contraste. En particulier, son objectif est de décrire les échanges entre le milieu sanguin et l'EEE dans le but de caractériser la microvascularisation des tumeurs.

Un protocole typique d'IRM-dynamique comporte une série d'images pondérées en  $T_1$  (e.g. avec des valeurs de  $\alpha$  différentes) avant l'administration

de l'AC dans le but de cartographier la valeur de base de  $T_1$  (information équivalente à  $R_{1,0}$ ). Par la suite, une séquence d'images avec les mêmes paramètres et un  $\alpha$  fixe est produite. Les premières images de cette série, avant l'injection de l'AC, servent à fixer le niveau de base du signal et les images subséquentes serviront à suivre la concentration d'AC dans les tissus.

Cette concentration est déterminée à partir de l'Eq. [9]. On a que :

$$S(t)(1 - e^{-T_R/T_1} \cos(\alpha)) = S_0(1 - e^{-T_R/T_1}) \sin(\alpha) \quad [11]$$

$$S(t) - S_0 \sin(\alpha) = (S(t) \cos(\alpha) - S_0 \sin(\alpha)) e^{-T_R/T_1} \quad [12]$$

$$-\frac{T_R}{T_1} = \ln \left( \frac{S(t) - S_0 \sin(\alpha)}{S(t) \cos(\alpha) - S_0 \sin(\alpha)} \right) \quad [13]$$

$$R_1(t) = \frac{1}{T_R} \ln \left( \frac{S_0 \sin(\alpha) - S(t) \cos(\alpha)}{S_0 \sin(\alpha) - S(t)} \right) \quad [14]$$

avec l'Eq. [10]

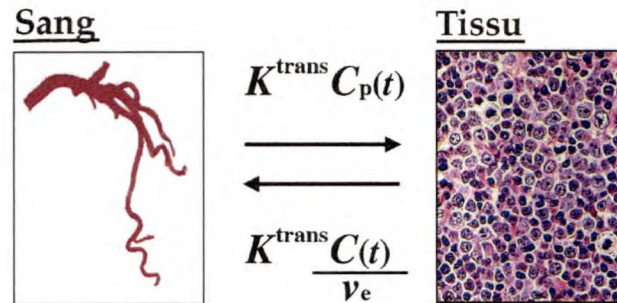
$$C(t) = \frac{1}{r_1} \left( \frac{1}{T_R} \ln \left( \frac{S_0 \sin(\alpha) - S(t) \cos(\alpha)}{S_0 \sin(\alpha) - S(t)} \right) - R_{1,0} \right) \quad [15]$$

Dans cette équation,  $S_0$  est déterminé à partir du signal de la première image sans AC ( $S(0)$ ), de la carte de  $T_1$  et de l'équation suivante tirée de l'Eq. [9]:

$$S_0 = S(0) \frac{1 - e^{-T_R/T_1} \cos(\alpha)}{(1 - e^{-T_R/T_1}) \sin(\alpha)} \quad [16]$$

## Modélisation pharmacocinétique

La modélisation pharmacocinétique appliquée à l'IRM-dynamique avec le Gd-DTPA tente de décrire l'état de la vascularisation dans les tissus en modélisant les échanges entre le sang et l'EEE. En effet, l'absence d'interactions biochimiques permet de décrire le comportement de cet AC à l'aide de modèles à deux compartiments (Fig. 3).



**Figure 3 :** Modèle à deux compartiments. L'AC passe de la circulation sanguine à l'EEE du tissu et vice versa. Les échanges entre le plasma sanguin et l'EEE sont déterminés par leur concentration respective ( $C_p(t)$  et  $C(t)$  [mM]) ainsi que par le taux de transfert transcapillaire ( $K^{\text{trans}}$  [ $\text{min}^{-1}$ ]) et la fraction de volume extravasculaire extracellulaire ( $v_e$ ).

Plusieurs modèles ont été développés depuis les débuts de l'IRM-dynamique (LARSSON *et al.*, 1990) (BRIX *et al.*, 1991) (TOFTS et KERMODE, 1991) (TOFTS *et al.*, 1999) (YANKEELOV *et al.*, 2005). Dans la majorité, les modèles utilisent les hypothèses suivantes (TOFTS, 1997) :

1. Les tissus peuvent être divisés en compartiments. Les caractéristiques de ces derniers ne changent pas au court d'une séance d'imagerie. La



concentration de l'AC est uniforme dans ces compartiments (i.e. mélange instantané du contenu des compartiments).

2. Le flux entre deux compartiments est proportionnel à la différence de concentration entre ceux-ci.
3. Les paramètres décrivant les échanges d'AC sont invariables durant une expérience donnée à un temps donné.
4. L'un des compartiments est le plasma sanguin avec une concentration  $C_p(t)$  en mM.
5. L'autre compartiment est l'EEE. Ce compartiment fait partie des tissus qui sont caractérisés par une concentration  $C(t)$  en mM et une fraction de volume extravasculaire extracellulaire ( $v_e$ ) sans unité.
6. La relaxivité de l'AC, en  $s^{-1} \text{ mM}^{-1}$ , est constante et correspond à la relaxivité in vitro de l'agent. Les effets sur le temps de relaxation  $T_1$  sont donnés par l'Eq. [10].
7. Toutes les molécules d'eau dans un compartiment perçoivent les effets de l'agent de contraste. Ainsi, durant un temps de répétition ( $T_R$ ), l'eau à l'intérieur des cellules, où le Gd-DTPA ne se rend pas (TOFTS, 1997), a suffisamment de temps pour passer au compartiment extravasculaire extracellulaire et voir sa magnétisation affectée par l'AC. Cette hypothèse porte le nom de "Fast Exchange Regime" dans la littérature

(YANKEELOV *et al.*, 2003). Certains modèles ont été publiés dans lesquels, cette hypothèse n'était pas utilisée et où le signal provenant des molécules d'eau à l'intérieur du milieu cellulaire avait une contribution distincte de celui provenant du compartiment où se loge l'AC (YANKEELOV *et al.*, 2003) (LI *et al.*, 2005).

En 1999, Tofts *et al.* ont publié une revue de littérature dans laquelle ils proposaient un modèle généralisé qui englobait plusieurs modèles utilisés à ce moment (TOFTS *et al.*, 1999). Ces travaux succédaient à une première revue de littérature (TOFTS, 1997) qui voulait rendre la notation uniforme pour faciliter la comparaison entre les travaux précédents et fonder une base commune pour les publications futures. Avant ces travaux, chaque modèle avait ses hypothèses sur les mécanismes régissant les échanges d'AC. Les équations suivantes montrent quelques unes de ces idées :

$$\text{Modèle limité par le flux sanguin} \quad \frac{d}{dt} C(t) = F \rho (1 - Hct) \left( C_p(t) - \frac{C(t)}{v_e} \right) \quad [17]$$

où  $F$  est la perfusion par unité de masse de tissu [ $\text{ml g}^{-1} \text{ min}^{-1}$ ],  $\rho$  est la densité du tissu [ $\text{g ml}^{-1}$ ],  $Hct$  est l'hématocrite.

$$\text{Modèle limité par la perméabilité des capillaires} \quad \frac{d}{dt} C(t) = PS \rho \left( C_p(t) - \frac{C(t)}{v_e} \right) \quad [18]$$

où  $PS$  est le produit de la perméabilité des parois capillaires sanguins et de leur surface par unité de masse [ $\text{cm}^3 \text{ min}^{-1} \text{ g}^{-1}$ ].

$$\text{Modèle combiné} \quad \frac{d}{dt}C(t) = EF\rho(1-Hct)\left(C_p(t) - \frac{C(t)}{v_e}\right) \quad [19]$$

où  $E$  est la fraction relative d'extraction entre le sang artériel et veineux (TOFTS *et al.*, 1999).

Le modèle que Tofts *et al.* proposent (Fig. 3) comporte un taux de transfert transcapillaire,  $K^{\text{trans}}$ , qui inclut les limitations dues au flux sanguin et à la perméabilité des capillaires sans faire la distinction entre celles-ci. Ainsi :

$$\text{Modèle généralisé} \quad \frac{d}{dt}C(t) = K^{\text{trans}}\left(C_p(t) - \frac{C(t)}{v_e}\right) \quad [20]$$

En solutionnant cette équation différentielle, on obtient une équation qui décrit la concentration de l'AC dans le tissu en fonction du temps et de  $C_p(t)$  :

$$C(T) = K^{\text{trans}} \int_0^T C_p(t) e^{-\frac{K^{\text{trans}}}{v_e}(T-t)} dt \quad [21]$$

Ainsi en ajustant les valeurs expérimentales de  $C(T)$  tirées de l'Eq. [15] avec la solution précédente, on obtient une valeur pour  $K^{\text{trans}}$  et  $v_e$ .

Bien entendu cette analyse sous-entend que la concentration dans le plasma sanguin peut être déterminée tout au long de l'expérience. Dans la littérature,  $C_p(t)$  est appelé la fonction d'entrée artérielle ("Arterial input function, AIF"). Prétendre connaître cette fonction n'est pas une affirmation triviale puisque plusieurs études ont montré les difficultés associées à son évaluation (PORT *et al.*, 2001) (YANKEELOV *et al.*, 2005). En effet, il est

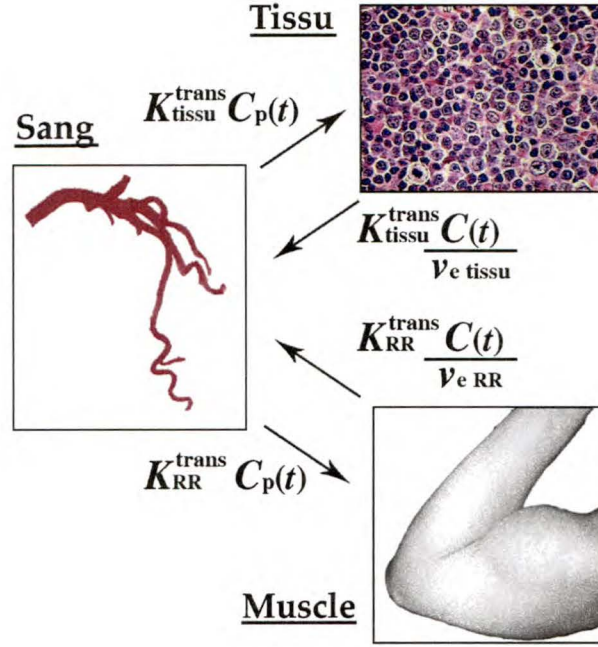
difficile de trouver un voxel d'une image d'IRM-dynamique qui puisse être associé uniquement à du sang artériel à cause des effets de volume partiel. Ce problème est d'autant plus gênant que le sujet imagé est petit. Ainsi pour l'étude de tumeurs chez le petit animal, d'autres recours doivent être utilisées. Certaines approches décrivent l'utilisation d'un cathéter pour récolter du sang et en faire l'analyse ultérieurement (FRITZ-HANSEN *et al.*, 1996). Le point faible de cette méthode est son aspect invasif. D'autres ont proposé l'utilisation d'une fonction générique appliquée à tous les sujets, mais Port *et al.* ont conclu en 2001 que l'utilisation d'une telle fonction n'est pas appropriée pour décrire les caractéristiques de la vascularisation d'un individu, mais s'applique davantage à l'étude des caractéristiques d'une population (PORT *et al.*, 2001).

Dans un article paru en 2005, Yankeelov *et al.* proposent un modèle utilisant une région de référence pour contourner l'épineux problème de la détermination de  $C_p(t)$  (Fig. 4) (YANKEELOV *et al.*, 2005). Les Eqs. [22] à [25] résument l'essentiel de leur théorie.

$$\frac{d}{dt} C_{\text{tissu}}(t) = K_{\text{tissu}}^{\text{trans}} \left( C_p(t) - \frac{C_{\text{tissu}}(t)}{v_{e,\text{tissu}}} \right) \quad [22]$$

$$\frac{d}{dt} C_{\text{RR}}(t) = K_{\text{RR}}^{\text{trans}} \left( C_p(t) - \frac{C_{\text{RR}}(t)}{v_{e,\text{RR}}} \right) \quad [23]$$

où les notations tissu et RR sont associées au tissu étudié et à une région de référence, respectivement. Dans le cas présent, la région de référence est un muscle.



**Figure 4 :** Modèle à deux compartiments avec région de référence. Le signal tiré du muscle est utilisé comme référence pour contourner le problème de détermination de la concentration plasmatique ( $C_p(t)$ ).

En remplaçant  $C_p(t)$  dans l'Eq [22] par le contenu de l'Eq [23], on obtient :

$$\frac{d}{dt} C_{\text{tissu}}(t) = K_{\text{tissu}}^{\text{trans}} \left( \frac{1}{K_{\text{RR}}^{\text{trans}}} \left( \frac{dC_{\text{RR}}(t)}{dt} + K_{\text{RR}}^{\text{trans}} \frac{C_{\text{RR}}(t)}{v_{e,\text{RR}}} \right) - \frac{C_{\text{tissu}}(t)}{v_{e,\text{tissu}}} \right) \quad [24]$$

$$C_{\text{tissu}}(T) = \frac{K_{\text{tissu}}^{\text{trans}}}{K_{\text{RR}}^{\text{trans}}} C_{\text{RR}}(T) + \frac{K_{\text{tissu}}^{\text{trans}}}{K_{\text{RR}}^{\text{trans}}} \left( \frac{K_{\text{RR}}^{\text{trans}}}{v_{e,\text{RR}}} - \frac{K_{\text{tissu}}^{\text{trans}}}{v_{e,\text{tissu}}} \right) \int_0^T C_{\text{RR}}(t) e^{-\frac{K_{\text{tissu}}^{\text{trans}}}{v_{e,\text{tissu}}}(T-t)} dt \quad [25]$$

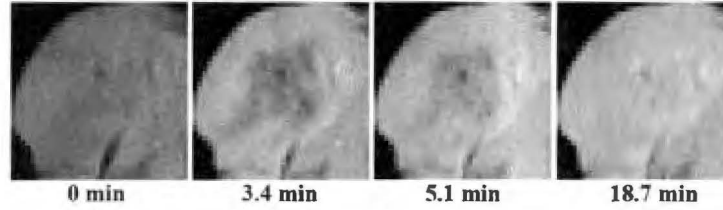
En utilisant cette dernière équation pour ajuster les données expérimentales de l'IRM-dynamique, on obtient des valeurs de  $K^{trans}$  et  $v_e$  pour les tissus étudiés sans avoir à déterminer la fonction d'entrée artérielle. Assurément, le choix du tissu de référence et celui des valeurs de  $K_{RR}^{trans}$  et  $v_{e,RR}$  ont un impact important sur les résultats tirés de la modélisation. En effet, les valeurs obtenues seront relatives à celles fixées pour le tissu de référence. Les valeurs qui ont été retenues pour ces paramètres sont  $K_{RR}^{trans} = 0.1 \text{ min}^{-1}$  et  $v_{e,RR} = 0.1$  (YANKEELOV *et al.* 2005).

## Modèle proposé

L'ensemble des modèles discutés dans la section précédente sont conçus pour une analyse pixel-par-pixel des images d'IRM-dynamique. Cette approche ne tient pas compte des échanges d'AC qui peuvent survenir par diffusion à l'intérieur des tissus. Pourtant l'hétérogénéité de la perfusion des tissus tumoraux qui a été mise en évidence par plusieurs études (JAIN, 1988) (ESKEY *et al.*, 1992) (SU *et al.*, 2003) (CHA *et al.*, 2003) engendre forcément des échanges d'AC à partir de régions bien perfusées vers d'autres moins bien vascularisées. La figure 5 montre une tumeur où ce phénomène est flagrant. En effet, le centre de celle-ci ne rehausse pas initialement lors des premiers

passages de l'AC et pourtant, après quelques minutes, il montre un rehaussement de son signal provenant de la périphérie bien perfusée.

L'ajustement de ces données avec un modèle standard et une approche pixel-par-pixel donnera des conclusions erronées sur l'état du micro-environnement tumoral (voir l'article : Incorporating Contrast Agent Diffusion into the Analysis of DCE-MRI Data. Pellerin M, Yankeelov TE et Lepage M, dans le chapitre Résultats).



**Figure 5 :** IRM-dynamique d'un adénocarcinome mammaire de souris. L'hétérogénéité de la distribution initiale de l'AC et le rehaussement graduel du centre de la tumeur montre la présence de diffusion de l'AC à l'intérieur du tissu.

Le modèle proposé dans ce mémoire inclut de façon explicite la diffusion de l'AC entre des pixels voisins. Ainsi, en ajoutant une part de diffusion au modèle standard de Tofts (Eq. [20]), on obtient :

$$\frac{d}{dt}C(\vec{r},t) = K^{\text{trans}}(\vec{r}) \left( C_p(t) - \frac{C(\vec{r},t)}{v_e(\vec{r})} \right) + \text{Diffusion}(\vec{r},t) \quad [26]$$

L'appendice de l'article : Incorporating Contrast Agent Diffusion into the Analysis of DCE-MRI Data. Pellerin M, Yankeelov TE et Lepage M cité en totalité ci-dessous montre que l'apport de la diffusion peut être approximé

par un échange moyen à travers chacune des surfaces d'un voxel. Cet échange ne dépend que du coefficient de diffusion  $D$  et du gradient de la concentration dans l'EEE entre deux voxels voisins.

$$\frac{d}{dt}C(\vec{r},t) = K^{\text{trans}}(\vec{r}) \left( C_p(t) - \frac{C(\vec{r},t)}{v_e(\vec{r})} \right) + \sum_{\text{voisins}} D(\vec{r}) \bar{\nabla} \left. \frac{C(\vec{r},t)}{v_e(\vec{r})} \right|_{\text{interface}} \frac{\bar{S}}{V} \quad [27]$$

où les valeurs de  $D$  et du gradient sont moyennées sur l'ensemble de la surface,  $\bar{S}$  est la surface orientée d'un des côtés du voxel et  $V$  le volume du voxel. Cette équation devient, en ne considérant que les échanges en deux dimensions dans le plan des images :

$$\begin{aligned} \frac{d}{dt}C_{i,j}(t) = & K_{i,j}^{\text{trans}} C_p(t) - K_{i,j}^{\text{trans}} \frac{C_{i,j}(t)}{v_{e i,j}} + \\ & + \frac{(D_{i,j} + D_{i,j+1})}{2} \frac{1}{a} \left( \frac{C_{i,j+1}(t)}{v_{e i,j+1}} - \frac{C_{i,j}(t)}{v_{e i,j}} \right) \frac{ae}{a^2e} \\ & + \frac{(D_{i,j} + D_{i,j-1})}{2} \frac{1}{a} \left( \frac{C_{i,j-1}(t)}{v_{e i,j-1}} - \frac{C_{i,j}(t)}{v_{e i,j}} \right) \frac{ae}{a^2e} \\ & + \frac{(D_{i,j} + D_{i+1,j})}{2} \frac{1}{a} \left( \frac{C_{i+1,j}(t)}{v_{e i+1,j}} - \frac{C_{i,j}(t)}{v_{e i,j}} \right) \frac{ae}{a^2e} \\ & + \frac{(D_{i,j} + D_{i-1,j})}{2} \frac{1}{a} \left( \frac{C_{i-1,j}(t)}{v_{e i-1,j}} - \frac{C_{i,j}(t)}{v_{e i,j}} \right) \frac{ae}{a^2e} \end{aligned} \quad [28]$$

Cette définition de la diffusion conserve la quantité d'agent de contraste. Dans la dernière équation, la notation est passée de la forme continue à celle discrète notée par les indices  $i$  et  $j$ . La concentration devient alors une matrice de format  $m \times n$  qui évoluera avec le temps. Les voxels sont, dans le cas



présent, des prismes à base carrée de côté  $a$  et d'épaisseur  $e$ . Les valeurs de  $m$ ,  $n$ ,  $a$  et  $e$  sont déterminées par les paramètres utilisés dans le protocole d'imagerie. La valeur moyenne du coefficient de diffusion pour chaque surface est prise comme la moyenne du coefficient de diffusion dans chacun des deux voxels adjacents. La définition actuelle du modèle comprend trois inconnus par voxel ( $K^{\text{trans}}$ ,  $v_e$  et  $D$ ). Dans le but de diminuer le nombre d'inconnus et donc la taille de l'espace des solutions, le coefficient de diffusion de l'AC sera posé comme étant le même que celui apparent de l'eau, connu pour chaque voxel à partir d'images pondérées par la diffusion (voir annexe A).

À l'aide de la transformation suivante :

$$\begin{pmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ C_{m1} & \dots & \dots & C_{mn} \end{pmatrix} \rightarrow \begin{pmatrix} C_{11} \\ C_{12} \\ \vdots \\ C_{1n} \\ C_{21} \\ \vdots \\ C_{mn} \end{pmatrix} \quad [29]$$

L'équation 27 devient :

$$\frac{d}{dt} \bar{\mathbf{C}}(t) = \bar{\mathbf{K}} \mathbf{C}_p(t) \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} - \bar{\mathbf{K}} \bar{\mathbf{V}} \bar{\mathbf{C}}(t) + \frac{1}{a^2} \bar{\mathbf{D}} \bar{\mathbf{V}} \bar{\mathbf{C}}(t) \quad [30]$$

Où  $\bar{K}, \bar{V}$  et  $\bar{D}$  sont des matrices  $mn \times mn$  et le vecteur rempli de 1 est de longueur  $mn$ .  $\bar{K}, \bar{V}$  sont des matrices diagonales ayant les valeurs de  $K^{trans}$  et  $1/v_e$  sur leur diagonal respectivement. La matrice  $\bar{D}$  est une matrice quasi-diagonale dont les éléments sont définis comme suit :

$$\begin{aligned} \bar{D}_{(i-1)*n+j, (i-1)*n+j} &= -\frac{1}{2} \left( (D_{i-1,j} + D_{i,j}) + (D_{i,j-1} + D_{i,j}) + \right. \\ &\quad \left. (D_{i,j+1} + D_{i,j}) + (D_{i+1,j} + D_{i,j}) \right) \\ \bar{D}_{(i-1)*n+j, (i-2)*n+j} &= \frac{D_{i-1,j} + D_{i,j}}{2} \quad , \quad \bar{D}_{(i-1)*n+j, (i-1)*n+j-1} = \frac{D_{i,j-1} + D_{i,j}}{2} \\ \bar{D}_{(i-1)*n+j, (i-1)*n+j+1} &= \frac{D_{i,j+1} + D_{i,j}}{2} \quad , \quad \bar{D}_{(i-1)*n+j, (i)*n+j} = \frac{D_{i+1,j} + D_{i,j}}{2} \end{aligned} \quad [31]$$

Le cas échéant, la contribution d'un voxel en dehors du tissu doit être retirée du terme de la diagonale de  $\bar{D}$  et le terme hors diagonale correspondant doit être mis à zéro. Ces critères posent les conditions aux limites permettant de conserver l'AC dans le tissu. Les définitions exactes de ces matrices sont reportées dans l'annexe B. Avec un pas de temps  $\Delta t$  assez court par rapport à celui durant lequel les molécules d'AC diffusent sur une distance moyenne de l'ordre de la dimension des voxels, l'approximation au premier ordre donne la solution suivante :

$$\bar{C}(t + \Delta t) = \Delta t \bar{K} C_{plasma}(t) \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} + \left[ \mathbf{I} + \frac{1}{a^2} \Delta t \bar{D} \bar{V} - \Delta t \bar{K} \bar{V} \right] \bar{C}(t) \quad [32]$$

Cette dernière équation est la formulation compacte du modèle proposé. Le terme  $\mathbf{I}$  représente la matrice identité de format  $mn \times mn$ . Dans le but de

diminuer la charge de calcul associée avec l'évaluation de  $\bar{C}(t)$  plusieurs fois, le modèle a été légèrement modifié pour diminuer le nombre de pas de calcul nécessaire pour conserver la validité de l'approximation au premier ordre de l'Eq. 31. Ainsi le modèle devient :

$$\bar{C}(t + \Delta t) = \Delta t \bar{K} C_{plasma}(t) \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} + \left[ \left( \mathbf{I} + \frac{1}{a^2} \Delta t' \bar{D} \bar{V} \right)^b - \Delta t \bar{K} \bar{V} \right] \bar{C}(t) \quad [33]$$

où  $\Delta t = b * \Delta t'$ . Cette version du modèle diminue la résolution temporelle des échanges entre le sang et le milieu interstitiel sans affecter celle du processus de diffusion. Le gain sur le temps de calcul provient du fait que le terme entre parenthèse ne sera évalué qu'une seule fois. Ainsi le nombre de pas de calcul est diminué d'un facteur  $b$ .

Malgré tout, la détermination des valeurs des composantes des matrices  $\bar{K}, \bar{V}$  demande passablement de puissance de calcul à cause de la taille de l'espace des solutions. En effet pour l'image d'une tumeur de  $25 \times 25$  pixels, le problème de trouver  $K^{trans}$  et  $v_e$  pour chacun des pixels consiste à trouver une solution idéale ayant  $2 \times 625$  paramètres. Un problème d'optimisation ayant un espace de solution de cette taille et qui contient d'innombrables minimums locaux requiert l'emploi de méthode d'optimisation stochastiques pour réussir à converger vers une solution acceptable. Deux de ces méthodes, les algorithmes génétiques et le recuit simulé, ont été implémentées et évaluées.

## **Algorithmes génétiques**

Les algorithmes génétiques (AG) font partie de la classe des méthodes d'optimisation aléatoires. Ils se basent sur les mécanismes de l'évolution pour engendrer des solutions  $S$ , appelées individus, de plus en plus adéquates à chaque nouvelle génération. Un bref aperçu du concept des AG sera présenté dans cette section. Pour plus d'information sur les AG, se référer à GOLDBERG, 1989.

Plutôt que de faire une recherche complètement aléatoire dans l'espace des solutions pour tenter de trouver la meilleure solution, les AG se servent des caractéristiques des individus les plus aptes d'une population pour créer une nouvelle génération d'individus qui s'approcheront davantage de l'extremum absolu de la fonction à optimiser. Ainsi, la recherche n'est pas parfaitement aléatoire, mais elle est plutôt dirigée par les succès et les échecs des générations précédentes qui se propagent dans le pool génétique d'une population.

L'un des principes à la base des AG est l'encodage des caractéristiques d'une solution de la même façon que les gènes humains sont codés dans l'ADN. Le codage le plus répandu est celui utilisant les nombres binaires pour représenter les différentes valeurs des paramètres d'une solution.

Les AG utilisent des mécanismes analogues à ceux de la reproduction sexuée pour combiner le bagage génétique de deux individus. Ainsi, dans le but de créer une nouvelle génération d'individus, des parents sont sélectionnés deux à deux pour créer un descendant qui aura une partie du bagage génétique de chacun de ses géniteurs. Le mécanisme qui crée les gènes du nouvel individu à partir de ceux des parents est appelé *crossover*.

Dans le but de créer des individus plus aptes, il doit y avoir une certaine forme de pression qui favorise certains individus par rapport aux autres. La fonction à optimiser  $F(S)$ , dorénavant appelée l'aptitude, joue ce rôle. Attention dans le cas présent,  $F(S)$  est une fonction à minimiser. Ainsi les individus les plus aptes auront des valeurs de  $F(S)$  minimales. Les individus sont ensuite classés en fonction de leur aptitude, ce qui détermine leur probabilité d'être sélectionné comme parent. Ainsi les caractéristiques des individus étant plus aptes ont plus de chance de demeurer dans le pool génétique de la prochaine génération que celles des individus médiocres. Cette forme de pression, oriente la recherche vers les solutions qui sont optimales.

Une opération essentielle au fonctionnement des algorithmes génétiques est celle de mutation. Dans le but d'assurer que toutes les solutions restent accessibles, il doit y avoir un mécanisme qui permet de conserver la

diversité génétique. En effet, certaines caractéristiques peuvent ne pas faire partie du pool génétique initial ou encore, elles peuvent avoir été retirées du pool d'une génération antérieure à cause qu'elles étaient cachées dans des individus médiocres qui n'ont pas engendré de descendants. Les mutations donnent la possibilité que certaines caractéristiques changent aléatoirement permettant de prévenir cette perte de diversité qui condamnerait les AG à trouver des solutions sous-optimales.

D'autres artifices peuvent être ajoutés aux AG pour améliorer leurs performances vis-à-vis de certaines classes de problèmes ou pour les empêcher de perdre trop de diversité génétique rapidement face à des supers individus qui dominent une population globalement médiocre. Entre autres, l'utilisation de l'élitisme pour faire passer le ou les meilleurs individus à la génération suivante permet de conserver la solution la plus apte trouvée durant toute l'exécution de l'algorithme.

## **Recuit simulé**

Le recuit simulé (RS, simulated annealing en anglais) fait aussi partie des outils de recherche stochastique. Cet algorithme est développé en analogie avec les méthodes de traitement thermique de métallurgie. Brièvement, dans

le cas d'un métal chauffé au-delà de son point de fusion, si la température est diminuée très lentement, le métal se solidifiera en minimisant l'énergie interne stockée dans l'arrangement de ses atomes. Il formera ainsi un cristal.

Pour l'optimisation par RS, un paramètre de contrôle,  $T$ , analogue à la température, est augmenté à un point où toutes les solutions de l'espace des solutions deviennent acceptables. Par la suite, la valeur du paramètre de contrôle est lentement diminuée et l'algorithme passe d'une solution à une autre dans son voisinage en suivant les règles du critère de *Metropolis* (METROPOLIS *et al.*, 1953). Le fonctionnement de l'algorithme sera brièvement abordé ci-dessous. Pour plus d'information, se référer à AARTS et KORST, 1989.

Dans le RS, le passage d'une solution  $S$  avec une aptitude  $F(S)$  vers une solution  $S'$  tirée du voisinage de la première suit les règles suivantes. Deux cas dictent la conduite de l'algorithme, soit  $F(S')$  est inférieure  $F(S)$  et le passage à la nouvelle solution est accepté, soit  $F(S')$  est supérieure à  $F(S)$  et le passage est accepté avec une probabilité,  $P(S, S', T)$ , donnée par le critère de *Metropolis* :

$$P(S, S', T) = e^{-\frac{(F(S') - F(S))}{T}} \quad [34]$$

en analogie avec la distribution de Boltzmann qui caractérise l'équilibre thermique d'un système :

$$P_i = \frac{1}{Z} e^{-\frac{E_i}{k_B T}} \quad [35]$$

où  $P_i$  donne la probabilité de trouver un système dans un état  $i$  caractérisé par une énergie  $E_i$ . Ici  $T$  est la température du système,  $k_B$  est la constante de Boltzmann et  $Z$  est donné par la somme sur tous les états  $j$  :

$$Z = \sum_j e^{-\frac{E_j}{k_B T}} \quad [36]$$

Dans le RS, plusieurs transitions doivent être effectuées avant que le paramètre de contrôle ne soit diminué à nouveau d'une petite quantité pour assurer que le système demeure à « l'équilibre thermique ».

Il a été démontré que l'algorithme de RS est formé d'une série de chaîne de Markov qui converge vers la solution optimale si la température descend infiniment lentement et qu'une infinité de transitions sont faites entre les baisses de température (AARTS et KORST, 1989). Bien entendu cette quantité astronomique de calculs n'a pas d'utilité, mais le RS reste une excellente méthode de recherche stochastique.



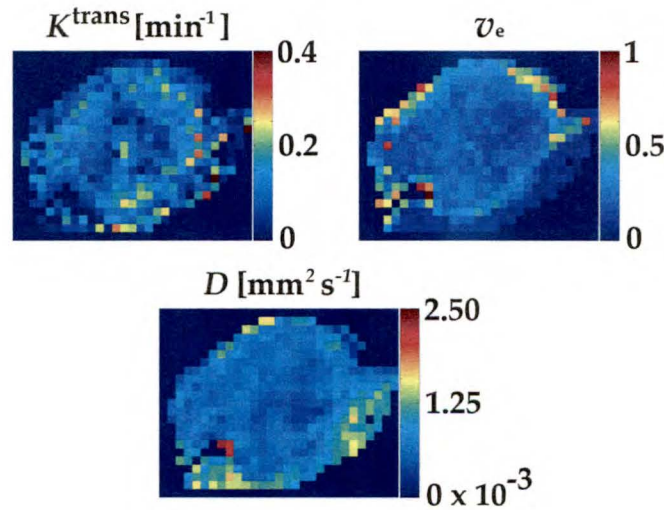
## MATÉRIELS ET MÉTHODES

---

### Simulation

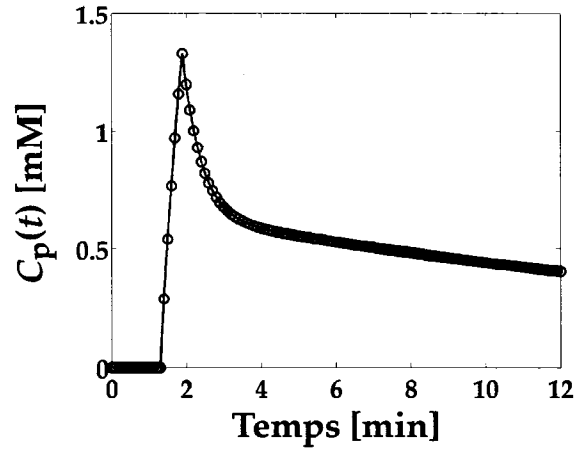
Des données simulées ont été utilisées pour évaluer les performances du modèle et des algorithmes d'optimisation utilisés tant qu'à leur reproductibilité, leur précision et leur robustesse face au bruit.

Ainsi des données ont été générées pour recréer les conditions dans lesquelles le modèle sera utilisé. La figure 6 montre les valeurs de  $K^{\text{trans}}$ ,  $v_e$  et  $D$  utilisées pour générer des images de concentration à l'aide du modèle proposé. La fonction d'entrée artérielle de la figure 7 a été créée par la convolution d'une fonction représentant une injection continue et uniforme pendant 30 s et d'une autre fonction représentant une élimination décrite par



**Figure 6 :** Paramètres utilisés pour créer les données simulées. a)  $K^{\text{trans}}$ , b)  $v_e$  et c)  $D$ .

une double exponentielle (WEINMANN *et al.*, 1984). Cette courbe est en accord avec celle donnée par PORT *et al.* en 2001.



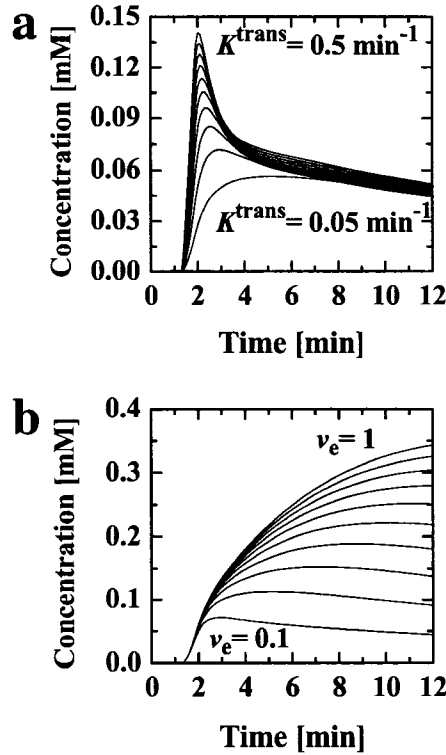
**Figure 7 :** Fonction d'entrée artérielle utilisée pour générer les données simulées.

La fonction,  $F(S)$ , utilisée pour évaluer l'aptitude d'une solution est donnée par:

$$F(S) = \sum_p A_p \frac{\sum_{i,j} (Données_{i,j,p} - Modèle_{i,j,p})^2}{\sum_{i,j} Données_{i,j,p}} \quad [37]$$

Dans cette équation,  $A_p$  est un facteur de poids qui accorde plus ou moins d'importance à différents moments de la série de données. Des calculs préliminaires, qui ne sont pas reportés dans ce texte, montraient que les algorithmes avaient de la difficulté à séparer le rôle de chaque paramètre et

avaient tendance à retourner des valeurs de  $K^{\text{trans}}$  trop faibles ( $K^{\text{trans}} \rightarrow 0 \text{ min}^{-1}$ ) et des valeurs de  $v_e$  trop élevées ( $v_e \rightarrow 1$ ). L'utilisation de la fonction ci-dessus permettait de mettre l'accent sur les rôles très différents de  $K^{\text{trans}}$  et de  $v_e$  de façon plus distincte. La figure 8 montre des courbes simulées à partir de la même fonction d'entrée artérielle (Fig. 7) pour un pixel sans diffusion (modèle généralisé de TOFTS *et al.* (TOFTS *et al.*, 1999)).



**Figure 8 :** Courbes simulées avec le modèle généralisé de Tofts *et al.* et la fonction d'entrée de la figure 7. a)  $K^{\text{trans}}$  varie de 0.05 à 0.50  $\text{min}^{-1}$  par incrément de 0.05 avec  $v_e = 0.10$ . b)  $v_e$  varie de 0.1 à 1.0 par incrément de 0.1 avec  $K^{\text{trans}}$  fixé à 0.1  $\text{min}^{-1}$ .

Dans cette figure, on remarque que les variations de  $K^{\text{trans}}$  ont beaucoup d'effet sur le comportement initial de la concentration dans le tissu. Le rôle de  $K^{\text{trans}}$  se reflète donc dans les premières images suite à l'injection de l'AC ( $t \approx 1.5$  min). De plus,  $v_e$  détermine le niveau de la concentration dans le tissu lorsque celle-ci atteint un équilibre avec le sang plusieurs minutes après l'injection de l'AC (dernières images). Ainsi la fonction utilisée pour calculer l'aptitude utilise  $A_p = 2$  pour les deux premières images après l'injection de l'AC et  $A_p = 1$  pour les deux dernières images de la série de données. Le choix de cette fonction aide les algorithmes à faire leur choix et biaise l'application du modèle. Il faut noter que de ne pas utiliser toutes les données de concentration empêche de diluer l'information des premières et dernières images dans l'ensemble, mais cela se fait au coût d'une robustesse diminuée face au bruit associé aux données.

### **Outils informatiques :**

L'ensemble des travaux de programmation a été effectué à l'aide de Matlab™.

Des exemples de codes sont reportés dans les annexes C et D.

Les calculs d'optimisation et de simulation ont été effectués sur différents ordinateurs :

- IBM ThinkPad™ : processeur Intel Pentium Mobile™ 1.6 GHz avec 512 Mb de RAM.
- Mammouth Série: 872 nœuds de 1 CPU Intel Pentium4™ 3.2 GHz avec 2Gb de RAM par nœud. ([www.ccs.usherbrooke.ca](http://www.ccs.usherbrooke.ca))

Les calculs effectués sur mammouth ont nécessité l'utilisation du compilateur de Matlab™ pour exporter les programmes codés avec Matlab™ sur les nœuds de calcul.

## **Optimisation par algorithmes génétiques**

L'utilisation des AG forme les premiers travaux d'optimisation sur l'application du modèle proposé. Certains raffinements du modèle n'ont pas été utilisés dans cette partie des travaux.

Une façon de coder les cartes de paramètres,  $K^{\text{trans}}$  et  $v_e$ , a dû être développée pour faire l'utilisation d'un AG pour l'optimisation. L'une des méthodes les plus utilisées reposent sur le codage des valeurs des paramètres en binaire. En effet l'utilisation d'une chaîne de "0" et "1" pour décrire un nombre permet de créer des « gènes » propices à l'application des différentes opérations de l'AG. Ainsi les différentes valeurs de  $K^{\text{trans}}$  ont été codées en binaire sur 6 bits [0 ,1 ,2 , ..., 63] . Ces valeurs étaient divisées par 100 avant

d'être utilisées par le modèle. Les valeurs discrètes de  $K^{\text{trans}}$  étaient donc [0, 0.01, 0.02, ... , 0.63]. De même  $v_e$  était codé sur 7 bits [0, 1, 2, ..., 127] et était divisé par 128 avant d'être introduit dans le modèle. La plage de 0 à 1 était couverte par incrément de 0.0078. Ce codage couvre une plage suffisamment grande et possède une précision adéquate comparée aux valeurs et aux erreurs rapportées dans la littérature pour le Gd-DTPA (PADHANI *et al.*, 2002) (DONAHUE *et al.*, 1995).

Le code mis en annexe permet de faire varier les différents paramètres de l'algorithme comme la taille de la population (PS), la probabilité de *crossover* (CP), la probabilité de mutation et le nombre d'individus élites qui passent à la génération suivante.

La population initiale était créée en assignant aléatoirement des valeurs de  $K^{\text{trans}}$  et de  $v_e$  pour chaque pixel associé au tissu. L'algorithme était biaisé par l'utilisation d'une distribution qui n'était pas parfaitement uniforme sur la plage de valeurs possibles pour ces deux paramètres. Ainsi une valeur moyenne ( $\langle K^{\text{trans}} \rangle = 0.1 \text{ min}^{-1}$ ,  $\langle v_e \rangle = 0.2$ ) et une déviation standard ( $\sigma K^{\text{trans}} = 0.05 \text{ min}^{-1}$ ,  $\sigma v_e = 0.05$ ) ont été utilisées pour aiguiller la recherche vers les régions de l'espace des solutions où l'aptitude serait hypothétiquement plus grande. Cette façon de faire n'empêche pas l'algorithme de trouver des

valeurs de paramètre écartées de ces valeurs moyennes à cause de l'introduction de mutations dans les gènes des nouveaux individus.

Les probabilités de sélection pour la reproduction étaient données par le rang de chaque individu classé selon son aptitude. Ainsi le dernier individu avait 1 part de l'ensemble, l'avant dernier 2 et le premier N parts où N est le nombre d'individu dans la population. Cette façon de faire, à l'opposé d'assigner une probabilité qui est fonction de l'aptitude, empêche un *super individu* de dominer la distribution de probabilité et d'avoir trop de descendants à son *image*. On peut ainsi conserver davantage de diversité génétique d'une génération à l'autre.

Les nouvelles générations sont créées de la façon suivante :

1. Sélection des individus les plus aptes qui passent directement à la prochaine génération (Élitisme).
2. Deux à la fois, des parents sont sélectionnés aléatoirement pour créer deux descendants. Les chances d'être choisi sont décrites par la distribution expliquée ci-dessus. Le bagage génétique des descendants est initialement la copie exacte de celui d'un des parents. Chaque gène à une probabilité de faire du *crossover* i.e. d'être sélectionné pour qu'une partie, choisie au hasard, soit échangée entre les deux descendants.

3. Chaque bit de chaque gène à une probabilité d'être muté, de passer de "0" à "1" ou de "1" à "0". Cette altération assure qu'un certain niveau de diversité génétique soit conservé.
4. L'aptitude de chaque descendant est évaluée. Dans le cas où l'un des deux descendants possède une aptitude anormalement élevée (5 fois plus que l'aptitude moyenne de la population précédente), les deux descendants sont rejetés.
5. Lorsque N nouveaux individus ont été créés (élitisme et reproduction confondus), la nouvelle génération est terminée. Recommencer à l'étape 1 pour une nouvelle génération.

En général, l'algorithme sera arrêté après un certain temps d'exécution. Le meilleur individu de la dernière génération correspond à la solution trouvée.

## **Optimisation par recuit simulé**

Pour l'optimisation par RS, les solutions ont été codées de la même manière que pour les AG. En effet, l'utilisation de valeurs discrètes pour les paramètres du modèle diminue énormément la taille de l'espace des solutions dans lequel s'effectue la recherche. De plus, avoir le même système de codage permettait de définir facilement le voisinage d'une solution. Ainsi, dans



l'algorithme, lorsque vient le temps de trouver une nouvelle solution dans le voisinage de celle déjà acceptée, on cherche une solution qui ne diffère que d'un bit dans tout le code génétique. Ainsi un seul paramètre change à la fois et il peut subir un changement de valeur plus ou moins important dépendamment de la position du bit modifié.

Dépendamment des données à ajuster, la température initiale sera fixée à un niveau permettant d'accepter la transition vers n'importe quelle solution. Ceci assure que l'algorithme ne sera pas piégé ou biaisé par la solution fixée initialement. L'un des moyens de fixer un niveau acceptable de température est décrit dans AARTS et KORTS, 1989. La méthode proposée augmente la valeur du paramètre de contrôle judicieusement jusqu'à ce que le rapport du nombre de transitions acceptées sur celui de celles proposées atteigne un pourcentage fixé par l'utilisateur, normalement près de 100%.

Le nombre de transitions qui doivent être effectuées entre chaque baisse de température a été fixé pour être de l'ordre de la taille du voisinage d'une solution (AARTS et KORTS, 1989). Ainsi pour un problème où la concentration dans le tissu est définie par une matrice de dimension  $10 \times 10$  avec 1 paramètre codé sur 6 bits et l'autre sur 7 bits, la longueur de la chaîne devrait être près de  $(7+6) \times 10 \times 10 = 1300$ . Si l'algorithme est lancé en parallèle

sur plusieurs machines, cette longueur peut être diminuée sans trop affecter la convergence vers la solution optimale (AARTS et KORST, 1989).

La chute en température qui a été utilisée est représentée par la fonction exprimée ci-dessous :

$$T_{n+1} = \varepsilon T_n \quad [38]$$

où  $T_n$  est la température actuelle,  $T_{n+1}$  celle de la prochaine série de transitions et  $\varepsilon$  est un paramètre qui régit la vitesse à laquelle la température chute.

Dans cette fonction  $\varepsilon$  doit être bien ajusté. En effet, l'utilisation d'une valeur de  $\varepsilon$  trop faible précipitera la chute de la température et l'algorithme convergera trop rapidement vers une solution sous-optimale. D'un autre côté, un  $\varepsilon$  trop élevé sera associé à une solution finale plus près de la solution optimale, mais demandera un temps de calcul trop élevé.

La meilleure solution trouvée au cours de l'exécution de l'algorithme doit être conservée en mémoire. Cette solution est celle qui est retenue à la fin pour évaluer les performances de l'algorithme.

## **Modèle animal**

Le modèle pharmacocinétique proposé a aussi été utilisé pour décrire des données réelles d'IRM-dynamique. Les tissus utilisés étaient des carcinomes

mammaires de souris (MC7-L1) (LANARI *et al.*, 2001). Les tumeurs étaient implantées chez des souris Balb-c femelles. Chez chaque animal,  $10^7$  cellules étaient injectées sous la peau au niveau de la hanche. Les séances d'imagerie ont été réalisées entre 4 et 8 semaines après l'implantation des tumeurs. Certaines tumeurs montraient des signes de nécrose.

## **IRM-dynamique**

Les travaux d'imagerie ont été réalisés à l'aide d'un appareil d'IRM de 7 Tesla fabriqué par la compagnie VARIAN. L'une ou l'autre des configurations suivantes a été utilisée : gradients 210/120 mm de  $30 \text{ G}\cdot\text{cm}^{-1}$  ou gradients 120/60 mm de  $100 \text{ G}\cdot\text{cm}^{-1}$  avec une sonde Millipede™ de 35 mm de champ de vue maximal.

Le protocole d'imagerie comportait les séquences suivantes :

1. Écho de gradient (Scout) :  $TR = 30 \text{ ms}$ ,  $TE = 2.3 \text{ ms}$ ,  $\alpha = 20^\circ$ ,  $FOV = 60 \text{ mm} \times 60 \text{ mm}$ , 3 tranches de 2 mm, résolution  $128 \times 128$ ,  $NA = 1$ . Une image prise dans chaque plan principal : axial, coronal, sagittal.
2. Écho de spin accéléré (Fsems) :  $TR = 2000 \text{ ms}$ ,  $TE = 12 \text{ ms}$ , nombre d'échos = 8,  $FOV = 32 \text{ mm} \times 32 \text{ mm}$ , 10 tranches de 1.5 mm, résolution  $128 \times 128$ ,  $NA = 1$ , axial.

3. Écho de gradient (Gems) :  $TR = \{100, 200\}$  ms,  $TE = 2.3$  ms,  $\alpha = \{10^\circ, 20^\circ, 25^\circ, 35^\circ, 50^\circ\}$ ,  $FOV = 32 \text{ mm} \times 32 \text{ mm}$ , 10 tranches de 1.5 mm, résolution  $128 \times 128$ ,  $NA = 4$ , axial.
4. Écho de gradient (Gems) :  $TR = \{100, 200\}$  ms,  $TE = 2.3$  ms,  $\alpha = 30^\circ$ ,  $FOV = 32 \text{ mm} \times 32 \text{ mm}$ , 10 tranches de 1.5 mm, résolution  $128 \times 128$ ,  $NA = 4$ , pour 40 images, axial.
5. Écho de spin pondéré par la diffusion (SemsDW) :  $TR = 2000$  ms,  $TE = 25$  ms,  $FOV = 32 \text{ mm} \times 32 \text{ mm}$ , 10 tranches de 1.5 mm, résolution  $128 \times 128$ ,  $NA = 1$ ,  $\delta = 3$  ms,  $\Delta = 9.5$  ms,  $G_{diff} = \{2, 10, 20, 30\} \text{ G} \cdot \text{cm}^{-1}$  dans la direction du *readout*, axial.

La première acquisition produit rapidement des images de basse qualité qui servent à la localisation de l'animal et au positionnement des images subséquentes. L'utilisation d'un *Fsems* sert au choix des tranches pour le reste de l'expérience. La première séquence de *Gems* sert à produire les images nécessaires à la création d'une carte des valeurs de  $T_1$  pour chacune des 10 tranches. La section de théorie décrit comment retrouver la valeur de  $T_1$  à partir d'acquisitions faites avec différentes valeurs de  $\alpha$ . La seconde séquence de *Gems* correspond à l'acquisition des images propres à l'IRM-Dynamique. Seule une partie des 40 images servira à l'application des modèles pharmacocinétiques. Les images *SemsDW* prises à la fin serviront à

déterminer le coefficient de diffusion apparent de l'eau, utilisé dans le modèle proposé.

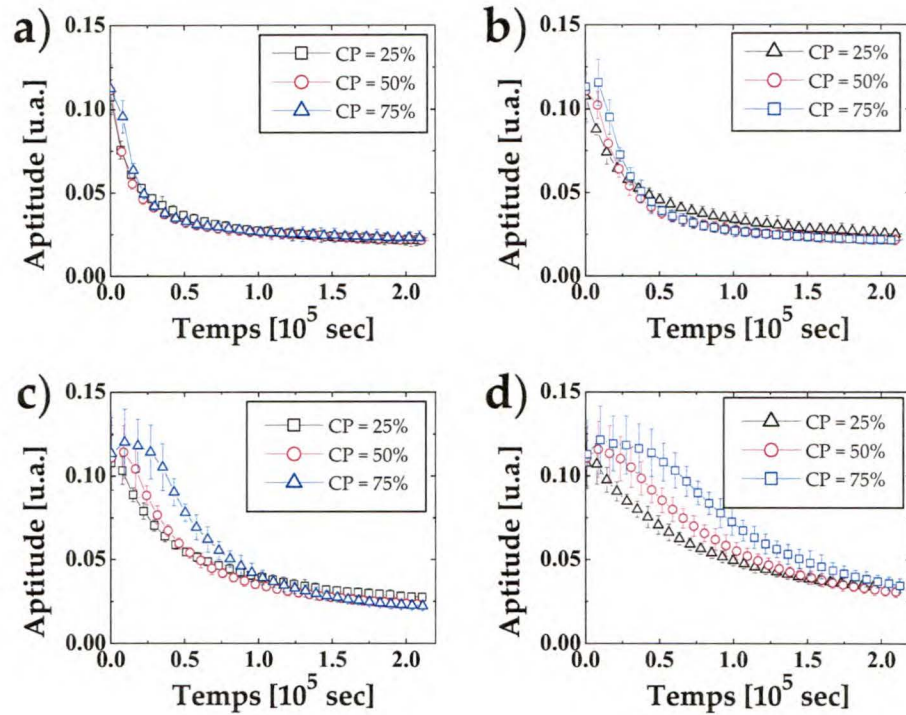
## RÉSULTATS ET DISCUSSIONS

---

### Algorithme génétique

Le comportement du modèle proposé couplé à l'utilisation d'un algorithme génétique a été étudié à l'aide des données simulées décrites ci-dessus. L'accent a été porté sur les effets de la taille de la population ainsi que sur les effets de la probabilité de *crossover*. Les autres paramètres comme la probabilité de mutation et l'élitisme assurent que l'algorithme ne fera pas d'erreurs irréparables, mais ils ne devraient pas être menés à dominer le comportement de l'algorithme.

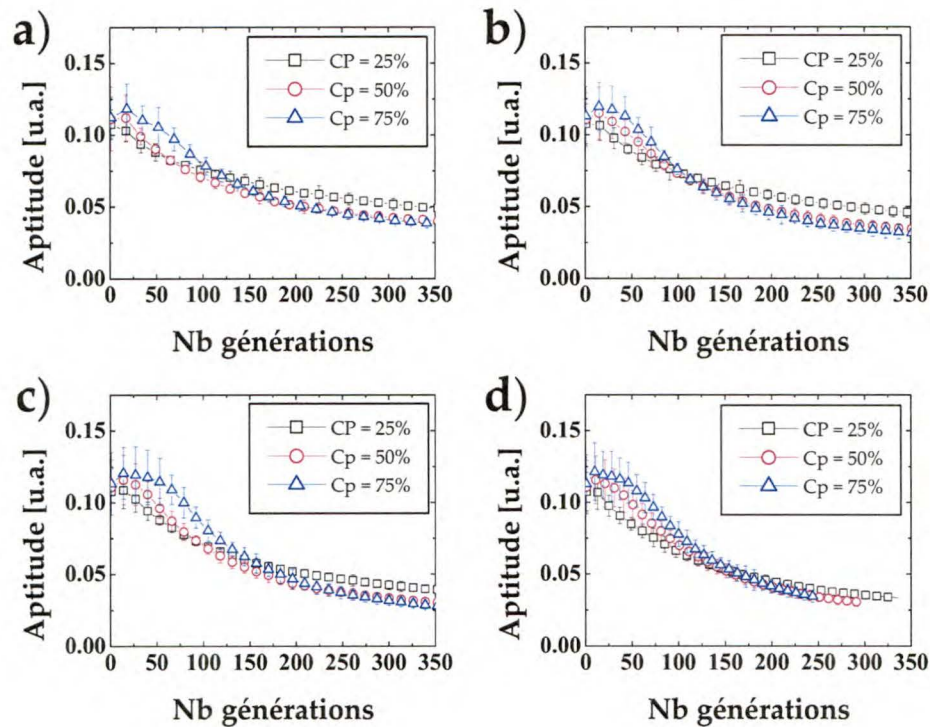
Ainsi différentes tailles de population ( $PS = \{500, 1000, 2000, 4000\}$  individus) et de probabilités de *crossover* ( $CP = \{25\%, 50\%, 75\%\}$ ) ont été utilisées pour définir les paramètres de l'AG. Chacune des combinaisons de ces paramètres a été utilisée trois fois pour confirmer la reproductibilité de l'algorithme. Le soin a été pris de s'assurer que le générateur de nombre aléatoire de chaque nœud produisait une série de nombres distincte pour chaque calcul. La figure 9 montre les performances de l'AG en fonction du temps de calcul pour les différentes tailles de population et probabilités de *crossover*.



**Figure 9 :** Aptitude de la meilleure solution en fonction du temps : a) Taille de la population (PS) = 500, b) PS = 1000, c) PS = 2000 et d) PS = 4000. Probabilité de mutation fixée à 0.05% et 2.5% des individus transférés de génération en génération par élitisme. Seule une fraction des données est montrée pour clarifier les graphiques.

De ces figures, on peut conclure que l'algorithme génère une recherche qui, bien que fondamentalement aléatoire, est reproductible tant qu'à l'aptitude de la meilleure solution au cours du temps. Pour chaque couple de paramètres (PS et CP), l'évolution de cette aptitude montre les mêmes structures : une descente exponentielle en deux parties. La descente initiale est liée au réarrangement des gènes contenus dans la population initiale. Par la suite, après avoir épuisé les possibilités du bagage génétique de cette population,

l'algorithme continue à progresser vers de meilleures solutions par l'application des mutations. À ce moment, l'algorithme devient similaire à une recherche purement aléatoire et ses performances chutent de façon importante.



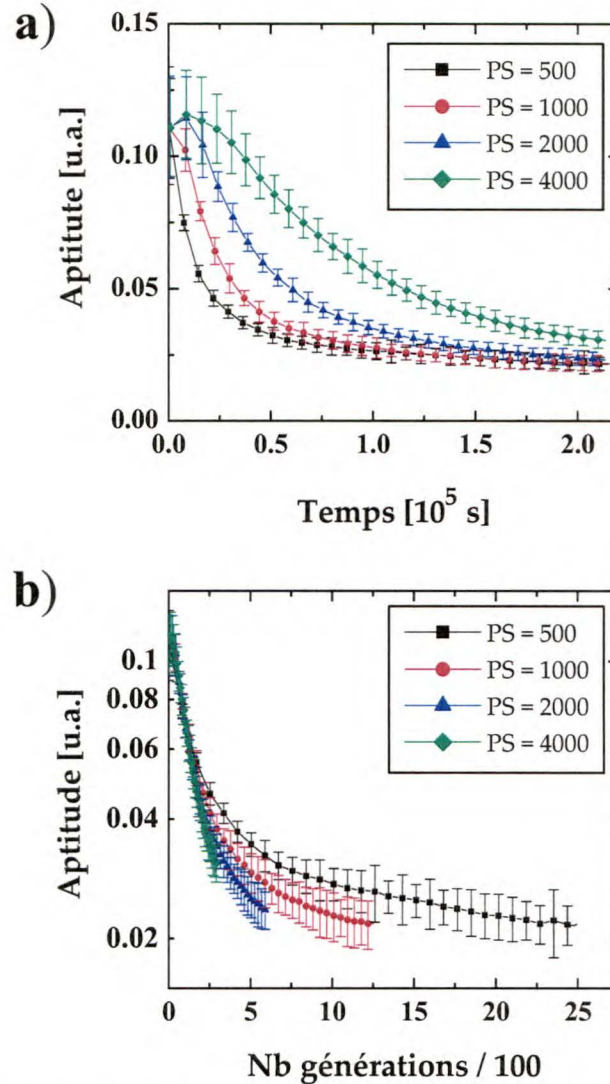
**Figure 10 :** Aptitude de la meilleure solution en fonction du nombre de générations complétées : a) Taille de la population (PS) = 500, b) PS = 1000, c) PS = 2000 et d) PS = 4000. Probabilité de mutation fixée à 0.05% et 2.5% des individus transférés de génération en génération par élitisme.

La figure 10 montre les mêmes résultats, mais en fonction du nombre de génération. La plage couverte par les axes des ordonnées dans cette figure est limitée par les calculs avec la population la plus grande, i.e. PS = 4000. En effet, le nombre de générations complétées dans un laps de temps dépend



linéairement de la taille de la population (e.g. pour 60 heures de calcul, 2500 générations complétées pour  $PS = 500$  alors que 300 générations complétées pour  $PS = 4000$ ) (voir Fig. 11 b). La figure 10 montre que CP a le même rôle peu importe la taille de la population. Si une faible probabilité de *crossover* semble augmenter les performances initiales de l'algorithme, elle mène aussi à la convergence vers des solutions légèrement moins bonnes (i.e. avec des valeurs d'aptitudes plus faibles). Dans tous les cas,  $CP = 50\%$  semble être un compromis adéquat.

Il est évident que la recherche est plus rapide lorsque la population est plus petite et que plus de générations peuvent être créées dans un même laps de temps. Cependant les calculs qui avaient de petites populations atteignaient rapidement la limite imposée par la taille de leur petit pool génétique (Fig. 11 a)). Comme il a été mentionné plus haut, la descente initiale est liée au réarrangement des gènes de la population, et la deuxième à une recherche aléatoire menée par l'introduction de mutations.



**Figure 11 :** Aptitude de la meilleure solution en fonction du temps (a) et du nombre de génération (b) pour l'optimisation par AG. Probabilité de *crossover* fixée à 50%, probabilité de mutation fixée à 0.05% et 2.5% des individus transférés de génération en génération par élitisme. Seule une fraction des données est montrée pour clarifier les graphiques.

Des résultats préliminaires avec le recuit simulé, qui ne sont pas reportés dans ce texte, ont montré que les AG n'étaient pas une méthode très efficace pour le

calcul en série. En effet, dans un laps de temps similaire, le RS atteignait des solutions ayant une aptitude inférieure par un ordre de grandeur à celle de la meilleure solution tirée des AG. Cette différence de performance a motivé l'abandon des AG pour ce problème d'optimisation.

Malgré tout, il faut se rappeler que ce qui limite grandement les performances des AG est le temps très long nécessaire à la création d'une nouvelle génération. En effet, la figure 11 b) montre que les calculs qui avaient une population de 4000 individus n'avaient pas exploité l'ensemble de la diversité génétique de la population initiale après 60 heures de calculs. L'une des façons qui permettrait de faire un grand bond dans la performance de l'algorithme serait de paralléliser les calculs pour diviser la lourde tâche de créer une nouvelle génération et d'évaluer l'aptitude de chacun des nouveaux individus entre plusieurs nœuds de calcul. De cette façon, le calcul serait accéléré d'un facteur égal au nombre de nœuds utilisés et l'AG pourrait utiliser des populations de grande taille.

## **Recuit simulé**

L'ensemble des travaux effectués sur l'utilisation du RS et les résultats associés sont reportés dans l'article inclus en totalité ci-dessous. Le RS s'est

avéré être la méthode d'optimisation la plus adéquate pour l'ajustement des données avec le modèle proposé. Dans l'article, se trouvent les travaux effectués sur des données simulées ainsi que sur des données expérimentales d'adénocarcinomes mammaires de souris.

# **Incorporating Contrast Agent Diffusion into the Analysis of DCE-MRI Data**

Martin Pellerin<sup>1</sup>, Thomas E. Yankeelov<sup>2</sup> and Martin Lepage<sup>1</sup>

<sup>1</sup> Centre d'imagerie moléculaire de Sherbrooke and Département de médecine nucléaire et de radiobiologie, Université de Sherbrooke, Sherbrooke, Québec, Canada. <sup>2</sup> Institute of imaging science, Vanderbilt University, Nashville, Tennessee, United States

**Submitted to Magnetic Resonance in Medicine on February 9<sup>th</sup> 2007**

## **ABSTRACT**

Standard two-compartment pharmacokinetic models that describe the DCE-MRI time course of Gd-DTPA concentration in a tissue do not account for the passive diffusion of contrast agent from a well perfused to a less vascularized region. The standard Tofts model returns inaccurate values of  $K^{\text{trans}}$  (mean absolute relative difference of 43%) from realistic simulated data where a sharp delineation exists between a well perfused and a poorly vascularized region. This contribution proposes a diffusion-perfusion (DP) model where contrast agent diffusion is explicitly taken into account and incorporated in the standard two-compartment Tofts model. The proposed diffusion-perfusion model reliably retrieved the values of  $K^{\text{trans}}$  and  $v_e$  (mean absolute relative difference of 16% and 17%, respectively) from simulated data. On mouse adenocarcinoma xenografts data showing evidence of contrast agent

diffusion, the standard model returned unphysical values of  $v_e$  in the tumor core whereas the proposed model found values that were in the physical range ( $0 < v_e < 1$ ) throughout the tissue. In addition,  $K^{\text{trans}}$  distributions from the proposed model were more closely corresponding to the observed sharp delineation between highly and poorly perfused areas in mouse tumors.

## INTRODUCTION

The exchange of an inert gas within a tissue was described in 1951 using a pharmacokinetic model to extract physiologically relevant parameters from time-dependent measurements of tracer concentration (1). Many authors have since adapted this early model to the specific requirements of dynamic contrast enhanced magnetic resonance imaging (DCE-MRI) experiments to relate the enhancement pattern of a tissue to the underlying exchange between blood plasma and the extravascular space (2-5). In the clinic, DCE-MRI has shown potential as a powerful tool to characterize the microvascular environment in tissue lesions as diverse as multiple sclerosis and tumors (6-7). Recent improvements to pharmacokinetic models accounted for additional phenomena, such as the variation in shutter speed limits associated with

contrast agent (CA) concentration (8-9). Those studies have shown that neglecting the limits of the transcytolemmal and transendothelial water exchanges can lead to erroneous estimations of standard model parameters. Recently, the tedious problem of evaluating the arterial input function (AIF) has been circumvented by the use of reference region models developed by several groups (10-14). To our knowledge, the possibility that a CA can passively diffuse from a well perfused region to a less vascularized region has not been addressed, although diffusion is known to occur within tumors (15).

Tumor tissues show clear signs of heterogeneity in their perfusion pattern (16-17). They are often characterized by a well perfused rim surrounding a poorly vascularized, possibly necrotic core (18-19). In DCE-MRI experiments, this translates into a rapid and intense signal enhancement at the rim followed by a delayed enhancement of the core, which may be the result of CA diffusion from the rim to the core. A common assumption made by all current models is that DCE-MRI data can be analyzed on a pixel-by-pixel basis, which inherently neglects CA diffusion. The mathematical models used with DCE-MRI experiments could potentially be improved in order to obtain a better representation of the underlying physiological processes. A better understanding of the microvascular environment of tumors could impact the treatment protocols using agents delivered *via* the

blood circulation, such as chemotherapy agents. In this case, high transcapillary exchange rates commonly found in tumors are expected to result in an increased concentration of the agent. In addition, anti-angiogenic treatments attempt to compromise the neovascular system (20-21), although an effective treatment may actually require the re-establishment of a functioning micro-vascular system (22). Finally, a high perfusion also correlates with a larger oxygen concentration in tumor tissue (23), which increases the radiosensitivity of a tumor (24).

In this contribution, we propose a diffusion-perfusion (DP) model where CA diffusion is explicitly taken into account and included in the standard Tofts model (6). Extensive characterization of the DP model is made using simulated data. The DP model is then applied to mouse data and the results are compared with those from the standard model.



## THEORY

### Proposed DP model:

Using the standardized symbolic notation described by Tofts *et al.* (6), the variation of CA concentration within a voxel is described by a standard two-compartment model such that:

$$\frac{d}{dt}C_{i,j}(t) = K_{i,j}^{\text{trans}}C_p(t) - K_{i,j}^{\text{trans}}\frac{C_{i,j}(t)}{v_{e\ i,j}} \quad [1]$$

where  $C_{i,j}(t)$  are the elements of the matrix  $\mathbf{C}$  representing the tissue concentration in mM,  $C_p(t)$  is the plasma concentration of CA in mM,  $K^{\text{trans}}$  is the transcapillary transfer rate in  $\text{min}^{-1}$  and  $v_e$  is the extravascular extracellular volume fraction. The value of  $i$  and  $j$  range from 1 to  $m$  and 1 to  $n$ , respectively where  $m$  is the number of frequency encoding steps and  $n$  is the number of phase encoding steps in the image acquisition matrix.

The variation of contrast agent concentration exchanged by diffusion from one voxel of dimension  $a \times a \times e$  to its immediate neighbor in 2D can be approximated by Eq. [2]:

$$\frac{d}{dt}C_{i,j}(t) = \sum_{\text{Neighbors}} \left( \frac{D_{i,j} + D_{\text{Neighbor}}}{2} \right) \frac{1}{a} \left( \frac{C_{\text{Neighbor}}(t)}{v_{e\ \text{Neighbor}}} - \frac{C_{i,j}(t)}{v_{e\ i,j}} \right) \frac{S}{V} \quad [2]$$

where  $\mathbf{D}$  is the matrix containing the diffusion coefficients of the CA for each voxel,  $V = a^2e$  is the volume of a voxel and  $S = ae$  is the surface between two

neighboring voxels. This description of CA diffusion assumes instantaneous mixing of the CA within a voxel and keeps the overall quantity of CA constant. Since the surface between voxels of adjacent slices is usually much smaller than between voxels of the same slice, the error associated to a 2D description of CA diffusion is small. A complete derivation of Eq.[2] can be found in the appendix.

Incorporating both the effects of active delivery (perfusion, Eq. [1]) with passive delivery (diffusion, Eq. [2]) yields the proposed DP model in 2D:

$$\frac{d}{dt}C_{i,j}(t) = K_{i,j}^{\text{trans}}C_p(t) - K_{i,j}^{\text{trans}}\frac{C_{i,j}(t)}{v_{e,i,j}} + \sum_{\text{Neighbors}} \left( \frac{D_{i,j} + D_N}{2} \right) \frac{1}{a} \left( \frac{C_N(t)}{v_{e,N}} - \frac{C_{i,j}(t)}{v_{e,i,j}} \right) \frac{S}{V} \quad [3]$$

where the subscript  $N$  stands for Neighbor. More explicitly, expanding Eq. [3] yields:

$$\begin{aligned} \frac{d}{dt}C_{i,j}(t) = & K_{i,j}^{\text{trans}}C_p(t) - K_{i,j}^{\text{trans}}\frac{C_{i,j}(t)}{v_{e,i,j}} \\ & - \frac{1}{a^2} \frac{D_{i+1,j} + D_{i-1,j} + D_{i,j+1} + D_{i,j-1} + 4D_{i,j}}{2} \frac{C_{i,j}(t)}{v_{e,i,j}} \\ & + \frac{1}{a^2} \frac{D_{i+1,j} + D_{i,j}}{2} \frac{C_{i+1,j}(t)}{v_{e,i+1,j}} + \frac{1}{a^2} \frac{D_{i-1,j} + D_{i,j}}{2} \frac{C_{i-1,j}(t)}{v_{e,i-1,j}} \\ & + \frac{1}{a^2} \frac{D_{i,j+1} + D_{i,j}}{2} \frac{C_{i,j+1}(t)}{v_{e,i,j+1}} + \frac{1}{a^2} \frac{D_{i,j-1} + D_{i,j}}{2} \frac{C_{i,j-1}(t)}{v_{e,i,j-1}} \end{aligned} \quad [4]$$

To simplify the notation and computation, the matrix  $\mathbf{C}$  of size  $m \times n$  is expressed as a vector  $\bar{\mathbf{C}}$  of length  $mn$  by the following transformation:

$$\mathbf{C}(t) = \begin{bmatrix} C_{1,1} & C_{1,2} & \cdots & C_{1,n} \\ C_{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ C_{m,1} & \cdots & \cdots & C_{m,n} \end{bmatrix} \rightarrow \bar{\mathbf{C}}(t) = \begin{bmatrix} C_{1,1} \\ C_{1,2} \\ \vdots \\ C_{2,1} \\ \vdots \\ C_{m,n} \end{bmatrix} \quad [5]$$

Equation [4] can be rewritten:

$$\frac{d}{dt} \bar{\mathbf{C}}(t) = \bar{\mathbf{K}} \mathbf{C}_p(t) \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} - \bar{\mathbf{K}} \bar{\mathbf{V}} \bar{\mathbf{C}}(t) + \frac{1}{a^2} \bar{\mathbf{D}} \bar{\mathbf{V}} \bar{\mathbf{C}}(t) \quad [6]$$

In this form,  $\bar{\mathbf{K}}$  and  $\bar{\mathbf{V}}$  are diagonal matrices of size  $mn \times mn$  with the values of  $K_{i,j}^{\text{trans}}$  and  $1/v_{e,i,j}$  on their diagonal, respectively. The column vector filled with ones has a length  $mn$ .  $\bar{\mathbf{D}}$  is a nearly diagonal sparse matrix of size  $mn \times mn$ .

From Eq. [4], the elements of  $\bar{\mathbf{D}}$  are given by:

$$\begin{aligned} \bar{D}_{(i-1)*n+j,(i-1)*n+j} &= -\frac{1}{2} \left( (D_{i-1,j} + D_{i,j}) + (D_{i,j-1} + D_{i,j}) + \right. \\ &\quad \left. (D_{i,j+1} + D_{i,j}) + (D_{i+1,j} + D_{i,j}) \right) \\ \bar{D}_{(i-1)*n+j,(i-2)*n+j} &= \frac{D_{i-1,j} + D_{i,j}}{2} \quad , \quad \bar{D}_{(i-1)*n+j,(i-1)*n+j-1} = \frac{D_{i,j-1} + D_{i,j}}{2} \\ \bar{D}_{(i-1)*n+j,(i-1)*n+j+1} &= \frac{D_{i,j+1} + D_{i,j}}{2} \quad , \quad \bar{D}_{(i-1)*n+j,(i)*n+j} = \frac{D_{i+1,j} + D_{i,j}}{2} \end{aligned} \quad [8]$$

Terms associated with a pixel located outside the tissue must be set to 0. This scheme prevents CA from leaking outside the tissue and so formulates the boundary conditions.

Using a short time step,  $\Delta t$ , during which only a small relative amount of CA is exchanged by diffusion between neighboring voxels, the first-order solution to Eq. [6] is:

$$\bar{\mathbf{C}}(t + \Delta t) = \Delta t \bar{\mathbf{K}} \mathbf{C}_p(t) \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \left( \mathbf{I} - \Delta t \bar{\mathbf{K}} \bar{\mathbf{V}} + \frac{\Delta t}{a^2} \bar{\mathbf{D}} \bar{\mathbf{V}} \right) \bar{\mathbf{C}}(t) \quad [9]$$

where  $\mathbf{I}$  stands for the identity matrix of size  $mn \times mn$ . This approximation is valid when the amount of CA transferred from one voxel to another by diffusion does not exceed a few percent per  $\Delta t$ .

The temporal resolution of conventional DCE-MRI images ranges from a few to tens of seconds. This resolution might be too low for the voxel size used in small animal scanners. For example, given a diffusion coefficient of  $1 \times 10^{-3} \text{ mm}^2/\text{s}$  and  $t = 60 \text{ s}$ , the average displacement of a free particle in 2D is  $\sqrt{4Dt} \approx 0.5 \text{ mm}$ . This is larger than the voxel dimension ( $\approx 0.25 \text{ mm}$ ) in these scanners, and this requires additional “diffusion steps” to be calculated at a time interval shorter than the experimental temporal resolution. However, a smaller  $\Delta t$  implies more computation steps and this lengthens the evaluation of  $\bar{\mathbf{C}}(t)$ . As a compromise, ten “CA exchange steps” between the plasma compartment and the extravascular extracellular space are considered between two successive images and the time resolution of the “diffusion

steps" is set close to 1 s. For example, if images are acquired with a temporal resolution of 50 s, exchange of CA is computed every 5 s and the diffusion of CA is computed every 1 s. The arterial input function (AIF) is interpolated to match the "exchange steps" time resolution (e.g., 5 s in the example). In this case Eq. [9] takes the form:

$$\bar{\mathbf{C}}(t + \Delta t) = \Delta t \bar{\mathbf{K}} C_p(t) \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \left( \left( \mathbf{I} + \frac{\Delta t'}{a^2} \bar{\mathbf{D}} \bar{\mathbf{V}} \right)^b - \Delta t \bar{\mathbf{K}} \bar{\mathbf{V}} \right) \bar{\mathbf{C}}(t) \quad [10]$$

where  $\Delta t' = \Delta t/b$  and  $b$  is an integer set so that  $\Delta t'$  is closest to 1 s. These additional diffusion steps do not increase considerably the overall computation time since the term to the power of  $b$  only needs to be evaluated once.

This model, as it stands, has three parameters for every voxel:  $K^{\text{trans}}$ ,  $v_e$  and  $D$ . To reduce the size of the search space and to prevent over-parameterization, the assumption that  $D$  is known for every voxel is made. As a first approximation,  $D$  is set equal to the apparent diffusion coefficient (ADC) of water as determined from diffusion weighted spin-echo images.

Clearly, a pixel-by-pixel fit is no longer justified since inter-voxel exchange of CA through diffusion requires fitting all voxels simultaneously. This increases the complexity of the fitting problem. For example, if the

values of  $K^{\text{trans}}$  and  $v_e$  are each discretized over  $2^7 = 128$  values, each pixel has  $2^{14}$  different solutions. For a tumor covering  $25 \times 25$  pixels, then a pixel-by-pixel approach needs to find the best among the  $2^{14}$  solutions for 625 uncoupled problems but the DP model needs to find the best among the  $(2^{14})^{625}$  solutions.

## **MATERIAL AND METHODS**

### **Optimization**

When considering the diffusion of CA between voxels, a complete image must be fitted as a whole using a 2D CA diffusion scheme, thereby increasing the complexity of the optimization problem. Furthermore, the “landscape” of the solution space is filled with countless local minima which prevent the use of standard techniques (e.g., steepest descent). An algorithm based on stochastic search methods is essential to efficiently converge to the optimal solution.

In the present study, the DP model is coupled to a simulated annealing search algorithm (see Ref. 25). Genetic algorithms were also tested but were found to be inefficient when using serial computing. The simulated annealing

algorithm searches randomly through the solution space, hopping from one solution to another using the Metropolis criteria (26). Given a function to be minimized,  $F(S)$ , henceforth called fitness, and  $T$  a control parameter, the transition from the current solution  $S_i$  to a randomly chosen neighboring  $S_i'$  is ruled as follows: if  $F(S_i') < F(S_i)$  then  $S_i'$  becomes the current solution, otherwise  $S_i'$  has a probability of becoming the current solution given by  $P(S_i, S_i', T) = \exp(-(F(S_i') - F(S_i))/T)$ . As will be detailed below, convergence is attained by lowering in steps the control parameter  $T$  after a fixed number of transitions have occurred. If  $T$  is lowered slowly enough, the algorithm will converge to a solution close to the absolute minimum. This search method is heuristic since the true solution is approached more closely as the search time increases.

The size of the solution space can be decreased by an appropriate discretization of the possible parameter values. The values of  $K^{\text{trans}}$  are coded in binary over 6 bits ( $K^{\text{trans}} \in \{0, 1, 2, \dots, 63\} / 100 \text{ [min}^{-1}\text{]}$ ) and  $v_e$  over 7 bits ( $v_e \in \{0, 1, 2, \dots, 127\} / 128$ ). This range of values encompasses those reported by other groups, and the interval between two values is smaller than errors reported (27-28). The neighborhood of a solution is chosen to include the solutions differing by only one bit in one parameter in only one pixel. The

number of transitions needed before stepping down the control parameter is fixed to half the size of the neighborhood as suggested in Ref. 25.

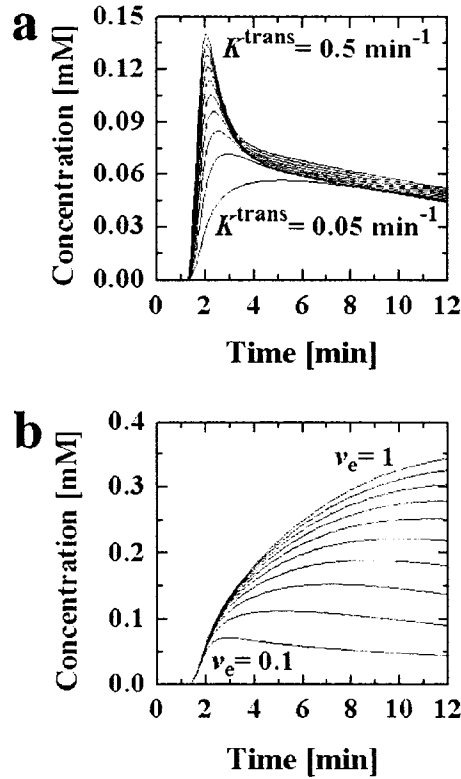
The descent of the control parameter  $T$  has the form:  $T(n+1) = \varepsilon T(n)$  where  $\varepsilon$  is a parameter associated to the speed of the descent and is in the range  $[0,1]$ . The initial value of  $T$  is set to allow the transition to any solution, better or worse. The main advantage of this approach is that the algorithm becomes insensitive to initial guesses.

The choice of the fitness function has a critical importance to determine the direction of the search. We found a standard sum of squared differences leads to erroneous solutions with too low values of  $K^{\text{trans}}$  and too high values of  $v_e$ . This problem is solved by considering only the part of the CA time course most sensitive to variations in  $K^{\text{trans}}$  and  $v_e$ . Figure 1 shows that the effect of varying  $K^{\text{trans}}$  is most significant in the initial uptake of CA. On the other hand, the effect of varying  $v_e$  is found in the “steady state” and “washout” occurring later after CA injection. Similar conclusions were reached in Ref. 29. The fitness function used for the simulations and real data fits is thus chosen to be:

$$F(S) = \sum_p A_p \frac{\sum_{i,j} (Model_{i,j,p} - Data_{i,j,p})^2}{\sum_{i,j} Data_{i,j,p}} \quad [11]$$



where  $S$  is the solution to be evaluated,  $A_p$  is a weighting factor and  $p$  ranges from one to the number of images. In all cases,  $A_p = 2$  for the first two images after CA injection, 1 for the last two images and 0 otherwise. This function



**Fig. 1:** Simulated data show the influence of  $K^{\text{trans}}$  and  $v_e$  on the shape of the CA concentration time course. **a:**  $K^{\text{trans}}$  values range from  $0.05 \text{ min}^{-1}$  to  $0.50 \text{ min}^{-1}$  by  $0.05$  increments while  $v_e$  is set to  $0.1$ . **b:**  $v_e$  values range from  $0.1$  to  $1.0$  by  $0.1$  increments while  $K^{\text{trans}}$  is set to  $0.1 \text{ min}^{-1}$ . The curves were calculated using the standard Tofts model and a simulated AIF.

was found to be reliable in retrieving values of  $K^{\text{trans}}$  and  $v_e$  used to generate simulated data. The choice of using only part of the data to evaluate the

fitness of a solution increases the sensitivity to noise. However, this approach prevents the algorithm from converging prematurely to “wrong” answers.

## **Computation**

The model and the simulated annealing algorithm were coded in Matlab™, compiled using the Matlab™ compiler, and exported to computation nodes. Computation was executed on a fraction of the serial supercomputer from the Université de Sherbrooke (Mammoth-Série, 872 nodes, Intel P4 with 2GB RAM per node).

## **Simulated data**

The performance of the model and the algorithm were tested against simulated data. Tissue concentration time courses were calculated using simulated  $K^{\text{trans}}$ ,  $v_e$  and ADC maps as an input to the DP model (see Fig. 4 below). These values will be referred to as the “true values”. In this example, perfusion is present throughout the tumor such that the values of  $K^{\text{trans}}$  are not dramatically different between the rim and the core of the tumor. More pronounced topological differences will be treated below. An arterial input function (AIF) was created using a box-shaped CA plasmatic input convoluted with a biexponential clearance (30). Noise was added to the tissue

concentration and to the AIF with a standard deviation equal to a fraction {0%, 1%, 2.5%, and 5%} of the highest concentration in each compartment, respectively.

## **Animal model**

The proposed DP model was also tested against real tumor data taken from mouse breast carcinomas (MC7-L1) (31). Subcutaneous tumors were inoculated (injection of  $10^7$  cells) in two Balb/c mice on the hind limb. Imaging was performed between 4 and 8 weeks after implantation.

## **MRI protocol**

Imaging was performed with a Varian 7T small animal scanner using a 40-mm Millipede™ probe and using either sets of gradient coils (210/120 mm with 30 G/cm or 120/60 mm with 100 G/cm). Mice were anaesthetized using a mix of 1.5-2% isoflurane in oxygen with the flow set at 2 l/min. A precontrast  $T_1$ -map was calculated from a set of  $T_1$ -weighed multi-flip angle gradient echo images ( $TR = 100$  or  $200$  ms,  $TE = 2.4$  ms,  $\alpha = \{10^\circ, 20^\circ, 25^\circ, 30^\circ, 35^\circ, 50^\circ\}$ ,  $FOV = 32 \times 32$  mm<sup>2</sup>, data matrix =  $128 \times 128$ ,  $NA = 4$ , 10 slices, 1.5 mm slice thickness).

Forty consecutive sets of images were acquired with the same parameters and  $\alpha$  fixed at  $30^\circ$ . The time resolution of those images were 51 s and 102 s for TR = 100 ms and 200 ms, respectively. A bolus of Gd-DTPA (Magnevist<sup>TM</sup>, Berlex) was injected *i.v.* via the tail vein after the second set of images (1.3 mmol/kg). The first 13 sets were used for fitting with the DP model in order to reduce computation time, which is proportional to the number of images considered.

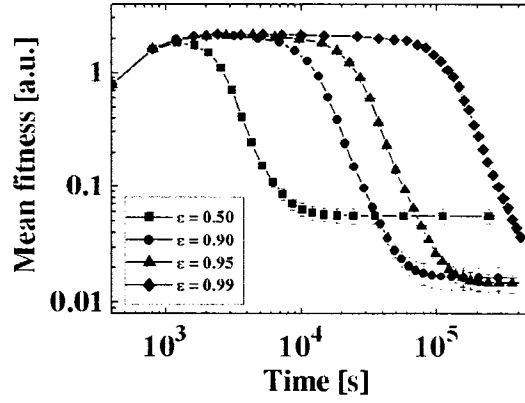
The ADC for each pixel was determined from a diffusion weighted spin-echo pulse sequence using b-values ranging from 2 to 492 s/mm<sup>2</sup> (TR = 2000 ms, TE = 25 ms, FOV = 32 × 32 mm<sup>2</sup>, data matrix = 128 × 128, NA = 1, 10 slices, 1.5 mm slice thickness,  $G_{\text{diff}} = \{2, 10, 20, 30\}$  G/cm in the readout direction,  $\delta = 3.5$  ms,  $\Delta = 9$  ms).

Studies have shown that the use of a generic AIF is not appropriate for individual estimation of CA uptake in tumors (32). Since it is difficult to accurately determine the AIF in individual animals, we derived the AIF from a reference region as suggested in Ref. 10 using  $K^{\text{trans,RR}} = 0.1 \text{ min}^{-1}$  and  $v_{e,RR} = 0.1$  (12). The standard Tofts model was used with the same reference region for comparison with the DP model on animal data (12).

## RESULTS AND DISCUSSION

### Simulations

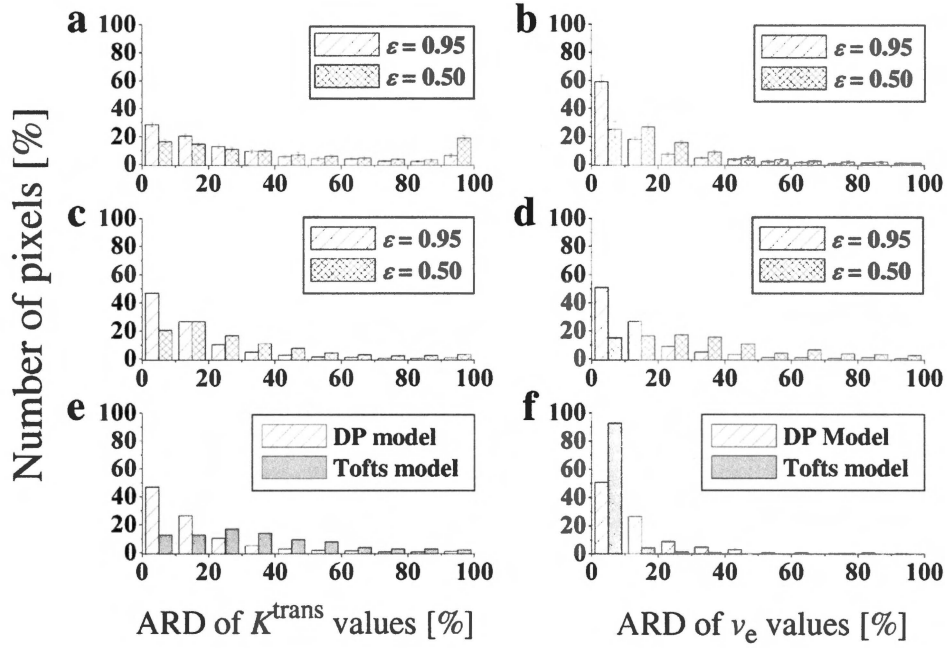
We evaluated the effect of changing the value of  $\varepsilon$  on the evolution of the fitness function during the fit. Figure 2 shows the value of the time variation of the mean fitness for four different values of  $\varepsilon$ . Only a fraction of the original dataset is displayed for clarity. Each point represents the mean values of 10 runs, every one of which converged to a different solution. Those results exhibit the characteristic behavior of heuristics; if the control parameter  $T$  is dropped too quickly (low value of  $\varepsilon$ ) the algorithm converged rapidly to a suboptimal solution. Figure 2 shows that the descent was too fast for  $\varepsilon = 0.5$  and the algorithm converged to a local minimum far from the optimal solution. A better solution (i.e., with a lower fitness) was obtained using a larger value of  $\varepsilon$  ( $\varepsilon = 0.95$ ). It is likely that the mean fitness would have been even lower for  $\varepsilon = 0.99$ , but convergence was not yet reached after 120 hours.



**Fig. 2:** Convergence of the DP model coupled to the simulated annealing algorithm fitting noiseless simulated data. The mean fitness is plotted as a function of fitting time for different values of  $\varepsilon$ . The initial rise found for all values of  $\varepsilon$  indicates the algorithm is not sensitive to the initial guess. The accuracy of the final solution is related to the mean fitness found for the lower plateau. More accurate values can be found for higher  $\varepsilon$ , at the expense of a larger computation time.

The initial guess was produced by a random distribution of  $K^{\text{trans}}$  and  $v_e$  values with a mean value of 0.10 min<sup>-1</sup> and 0.20 and a standard deviation of 0.05 min<sup>-1</sup> and 0.05, respectively. However, the initial rise of the mean fitness value demonstrates that the algorithm is insensitive to the initial guess since it originally accepts numerous poorer solutions.

The performance of the DP model was evaluated by computing the absolute relative difference (ARD) between fitted and true  $K^{\text{trans}}$  and  $v_e$  values (Fig. 3). The results for two different values of  $\varepsilon$  are displayed as histograms of the fraction of pixels and their ARD for both  $K^{\text{trans}}$  and  $v_e$  on Fig. 3a and Fig. 3b, respectively. It can be observed that the highest value of  $\varepsilon$  (the slowest



**Fig. 3:** Histograms of the number of pixels plotted against their absolute relative difference (ARD) to the true parameter values used to create the simulated concentration data. **a:**  $K^{\text{trans}}$  and **b:**  $v_e$  for a single run of the algorithm. **c:**  $K^{\text{trans}}$  and **d:**  $v_e$  for the mean solution taken from 10 runs. **e:**  $K^{\text{trans}}$  and **f:**  $v_e$  for both the DP model ( $\epsilon = 0.95$ , means of 10 runs) and the standard Tofts model.

descent) leads to more accurate solutions; i.e., more pixels have a smaller difference compared to the true values. Indeed for  $\epsilon = 0.95$ , a difference below 20% of the true value is found for 49% of the  $K^{\text{trans}}$  values and 77% of  $v_e$  values. Recalling that the increments in  $K^{\text{trans}}$  and  $v_e$  are of 0.01 and 0.0078, respectively, a difference of 10% from a value of 0.1 represents a difference close to a single increment. Nevertheless the accuracy of the solutions can be improved by averaging the results from multiple independent computations

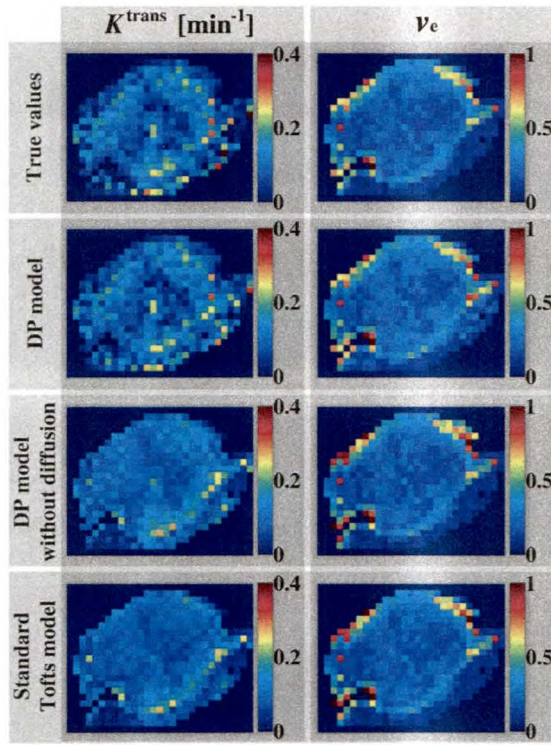
(Fig. 3c and Fig. 3d). Using 10 calculations and  $\varepsilon = 0.95$ , 73% of the  $K^{\text{trans}}$  values are now within 20% of the true value. However, it can be noticed that even if 77% of the  $v_e$  values are still within 20% of the true values, it appears that averaging somewhat reduced the accuracy of this parameter. Each individual run had a similarly low mean fitness, but  $K^{\text{trans}}$  and  $v_e$  were averaged on a pixel-by-pixel basis. Thus, if a particular pixel in any of the 10 runs did not converge to the true value, the average value for that pixel is likely to be affected. Overall, averaging multiple runs gives more accurate results and this approach is well suited for adapting simulated annealing to parallel computing (25).

Next, the performance of the DP model was compared to the standard Tofts model using the same simulated data for an average of 10 runs and  $\varepsilon = 0.95$ . Using the DP model, 73% of the  $K^{\text{trans}}$  and 77% of the  $v_e$  values were within 20% of the true values (mean ARD from the DP model is 16% for  $K^{\text{trans}}$  and 17% for  $v_e$ ), compared to 25% and 97% for the standard model (mean ARD from the standard model is 43% for  $K^{\text{trans}}$  and 4% for  $v_e$ ) (Fig. 3e and Fig. 3f). In a heterogeneous tissue, pixels with a high true value of  $K^{\text{trans}}$  receiving a large amount of CA lose some of their content to neighboring voxels with a lower value of  $K^{\text{trans}}$ , thus effectively smoothing the distribution of CA. As a consequence, the Tofts model is smoothing  $K^{\text{trans}}$  of these voxels to depict their



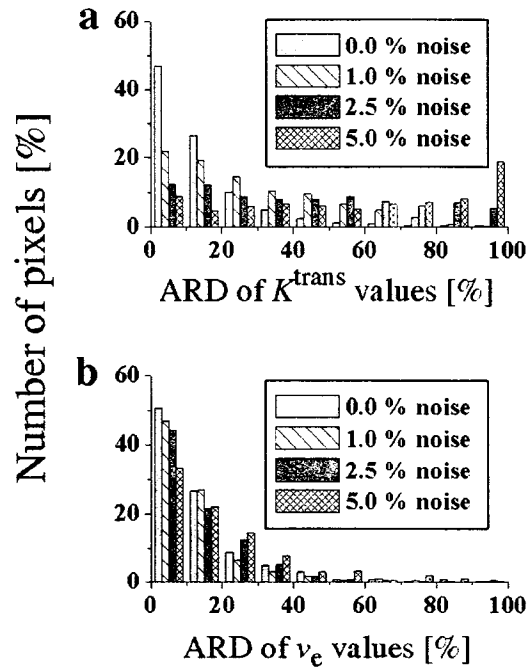
time courses. In other words, the Tofts model will find a lower  $K^{\text{trans}}$  for the voxels supplying CA to their neighbors and a higher  $K^{\text{trans}}$  for those being supplied. This can be seen on Fig. 4 where the  $K^{\text{trans}}$  map returned by the Tofts model is a smoothed version of the true distribution. Conversely, the DP model preserved the sharpness of the  $K^{\text{trans}}$  distribution. We recall that the values of  $K^{\text{trans}}$  were still relatively uniform throughout the tissue and there was no necrotic center, where the true value of  $K^{\text{trans}}$  would be approximately zero. In this case, the Tofts model yielded accurate values of  $v_e$ , which is mostly influenced by the “steady state” concentration found when the tissue reaches equilibrium with the blood plasma concentration several minutes after CA injection (see Fig. 1).

If the algorithm is run with the diffusion scheme turned off in the DP model (mean of 10 runs,  $\varepsilon = 0.95$ ), the mean values of  $K^{\text{trans}}$  and  $v_e$  obtained are similar to those from the standard Tofts model (Fig. 4) which indicates the Tofts model is a limit case of the DP model where  $D = 0$ .



**Fig. 4:** Results of the DP model (with and without the diffusion scheme,  $\varepsilon = 0.95$ , means of 10 runs) and the standard Tofts model. Results show that the simulated annealing algorithm coupled to the DP model can retrieve the true values of both parameters. The use of the standard Tofts model returns excellent maps of  $v_e$  but somewhat less accurate, smoothed values of  $K^{\text{trans}}$ . Removing the diffusion scheme from the DP model yields inaccurate values of  $K^{\text{trans}}$  similar to those from the standard model.

Figure 5 shows the effect of adding noise (0 to 5% of the maximum concentration) on the accuracy of the solutions returned by the DP model.  $K^{\text{trans}}$  is mostly influenced by the variations in concentration immediately after injection of CA (Fig. 1). Noise added to these concentrations is therefore likely to severely affect the fitted  $K^{\text{trans}}$  values. On the other hand,  $v_e$  regulates



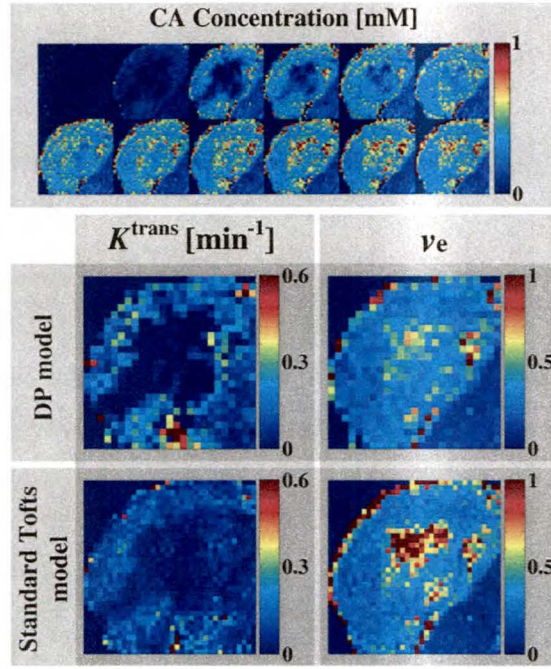
**Fig. 5:** Effect of noise (0 to 5% of the maximum concentration) on the accuracy of the parameter values returned by the DP model ( $\varepsilon = 0.95$ , means of 10 runs). Histograms show the number of pixels plotted against their absolute relative difference (ARD) to the true parameter values. Noise was added to both the simulated concentration data and the AIF.  $K^{\text{trans}}$  shows the greatest sensitivity to noise (a) while  $v_e$  is less affected (b).

the concentration reached in the “steady state” during which the concentration in the extravascular extracellular space follows the plasma concentration. It is not surprising then that this parameter is less sensitive to random variations around this slowly varying concentration. For comparison,

data acquired from mice with a 7T small animal scanner display a noise level lower than 1% of the maximum concentration.

## **Experimental Data**

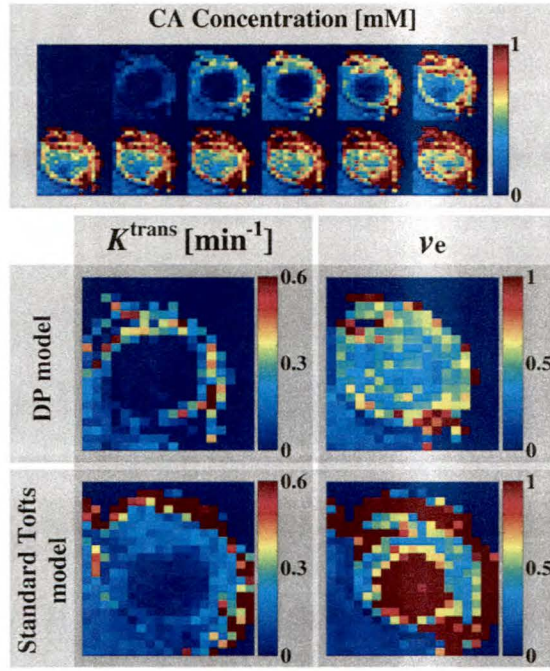
The top panels of Fig. 6 and Fig. 7 show examples of xenograft tumors displaying evidence of CA diffusion from the rim to the core. Both datasets were fitted with the DP model and the standard Tofts model using the same reference region (12). Since all pixels are fitted as a whole in the DP model, pixels having an odd behavior can hinder the convergence of the algorithm. Hence pixels with irregular CA concentration time courses, pixels showing partial filling effects or pixels for which no values of ADC could be calculated were removed from the datasets used with the DP model. Those pixels were mostly found on the edge of the tissue where the standard model returned unreasonably high values of  $K^{\text{trans}}$  and  $v_e$  (Fig. 7).



**Fig. 6: Top panel:** CA concentration calculated from DCE-MRI images of a mouse breast carcinoma xenograft. Imaging time resolution was 102 s. **Bottom panel:**  $K^{\text{trans}}$  and  $v_e$  maps returned by the DP model (means of 50 runs,  $\varepsilon = 0.9$ ) and the standard Tofts model. The data used by the DP model has been down sampled to reduce computation time. A sharp delineation observed in the dynamic images is clearly seen in the  $K^{\text{trans}}$  map from the DP model, but not from standard pixel-by-pixel approach.  $v_e$  values from the DP model are reasonable in the core of the tumor, as opposed to unphysical values from the standard model.

Looking at the images of the concentration as a function of time, a clear delineation is apparent between the rim and the core of the tumor. The concentration in the rim rises much more quickly than in the core. Those enhancement patterns have been reported by other groups (18-19) and have been correlated with microvessel density measurements (33). This delineation





**Fig. 7: Top panel:** CA concentration taken from DCE-MRI images of a mouse breast carcinoma xenograft. Imaging time resolution is 51 sec. **Bottom panel:**  $K^{\text{trans}}$  and  $v_e$  maps returned by the DP model (means of ten runs,  $\varepsilon = 0.9$ ) and the standard Tofts model. The standard model yielded unphysical  $v_e$  values ( $v_e > 1$ ) for 40% of the retained tumor pixels (excluding the rim) as opposed to the results from the DP model.

is not clear on the parameter maps from the standard Tofts model. However, the  $K^{\text{trans}}$  maps obtained from the DP model show the expected sharp distribution. This situation is similar to that discussed above for the simulated data.

For both datasets, the standard model returned unphysical values of  $v_e$  (i.e.,  $v_e > 1$ ) in the core of the tumors (reaching 40% of the retained pixels of the tumor in Fig. 7). This is because the slow uptake of CA by diffusion forced

the pixel-by-pixel approach to assign unrealistic  $v_e$  values to account for pixels that accumulate a significant amount of CA even though their  $K^{\text{trans}}$  value is negligible (34). The DP model returned more sensible values in the core of the tumors. While it is beyond the scope of this contribution to confirm these values correspond to the underlying physiology of the tumor, the fact that the values do not approach the maximum of the possible  $v_e$  values is certainly encouraging.

## CONCLUSIONS

We proposed a diffusion-perfusion pharmacokinetic model where diffusion of CA from one voxel to its neighbors was explicitly taken into account and incorporated into the standard Tofts model. We showed that the proposed DP model could reliably retrieve both  $K^{\text{trans}}$  and  $v_e$  from simulated data (mean ARD of 16% for  $K^{\text{trans}}$  and 17% for  $v_e$ ) even in the presence of realistic noise. When applied to the analysis of mice breast carcinoma xenograft data, the model outperformed the pixel-by-pixel analysis of the standard Tofts model by yielding sharper  $K^{\text{trans}}$  distributions in accordance with what could be observed in the time-varying concentration data. The DP model also yielded

$v_e$  maps that did not contain unrealistic values. This is in contrast to the standard model that cannot describe the slow uptake of CA in a poorly perfused, potentially necrotic, region.

This first attempt to incorporate CA diffusion into a pharmacokinetic model used several approximations and the following refinements will be examined. The diffusion coefficient was approximated to the ADC of water obtained from diffusion-weighted images. Although this approximation is not crude, a better correspondence will be needed between those two quantities. Furthermore, it has been shown that the ADC value of water in tissue is dependent on the range of b-values used in the imaging protocol (35). The model uses a 2D description of the diffusion process that can readily be expanded to 3D at the expense of computation time. One of the limitations of the DP model is the heavy computation power needed to carry out the stochastic search. The use of more adequate coding languages and refinement in the elaboration of the algorithm could increase the speed of the algorithm (Using C instead of Matlab<sup>TM</sup>). A thorough study of the best parameters (initial temperature, descent rate and neighborhood) that shape the simulated annealing algorithm could lead to more optimized use of the computation time. The form of the fitness function could also be altered to take advantage of the complete dataset.



Our results indicated a larger range of  $K^{\text{trans}}$  values was obtained by the DP model whereas the standard model appeared to smooth these values. Beyond providing potentially more physiologically relevant information on the tumor micro-environment, this larger range is anticipated to provide better diagnostic information based on this parameter. As for all other pharmacokinetic models, a histology-based validation is required.

## **ACKNOWLEDGEMENTS**

We wish to thank Mélanie Archambault for expert animal handling, Célène Dubuc, Simon Authier and Véronique Dumulon for the preparation of the animal models, Luc Tremblay for technical assistance and David Sénéchal for helpful discussions. This work was funded by the National Cancer Institute of Canada (grant number 016436). MP is supported by a student stipend from NSERC, TEY is supported by NIH/NIBIB 1 K25 EB005936-01, and ML is the Canada Research Chair in magnetic resonance imaging.

## APPENDIX

### Diffusion at the surface of a voxel

The variation of the amount of contrast agent  $Q(t)$  present in a voxel from diffusion to neighboring voxels can be described by the continuity equation:

$$\frac{d}{dt}Q(t) = \int_{\text{surface}} \vec{J}(\vec{r}, t) \cdot d\vec{s} \quad [\text{A1}]$$

where  $\vec{J}$  is the flux of contrast agent. The amount of CA is given by

$$Q(t) = \int_{\text{Volume}} C_e(\vec{r}, t) v_e(\vec{r}) dV = \int_{\text{Volume}} C(\vec{r}, t) dv = \langle C(\vec{r}, t) \rangle V \quad [\text{A2}]$$

where  $C_e(\vec{r}, t)$  and  $C(\vec{r}, t)$  are the CA concentration in the extravascular extracellular space and the CA concentration in the tissue, respectively, and  $V$  is the voxel volume. The concentrations are related by the extravascular extracellular volume fraction  $v_e$  such that  $C(\vec{r}, t) = C_e(\vec{r}, t) v_e(\vec{r})$ . The " $\langle \rangle$ " symbols express the mean value of the quantity. Equation [A1] becomes:

$$\frac{d}{dt} \langle C(\vec{r}, t) \rangle V = \int_{\text{surface}} \vec{J}(\vec{r}, t) \cdot d\vec{s} \quad [\text{A3}]$$

Applying Fick's first law of diffusion yields:

$$\frac{d}{dt} \langle C(\vec{r}, t) \rangle V = \int_{\text{surface}} D(\vec{r}) \vec{\nabla} C_e(\vec{r}, t) \cdot d\vec{s} = \int_{\text{surface}} D(\vec{r}) \vec{\nabla} \left( \frac{C(\vec{r}, t)}{v_e(\vec{r})} \right) \cdot d\vec{s} \quad [\text{A4}]$$

where  $D(\vec{r})$  is the diffusion coefficient of CA. The integral of the last equation can be calculated over the sub-surfaces separating one voxel from its neighbors. This leads to:

$$\frac{d}{dt}\langle C(\vec{r}, t) \rangle V = \sum_{Neighbors} \int_{Sub-surface} D(\vec{r}) \vec{\nabla} \left( \frac{C(\vec{r}, t)}{v_e(\vec{r})} \right) \cdot d\vec{s} \quad [A5]$$

$$\frac{d}{dt}\langle C(\vec{r}, t) \rangle V = \sum_{Neighbors} \left\langle D(\vec{r}) \vec{\nabla} \left( \frac{C(\vec{r}, t)}{v_e(\vec{r})} \right) \right\rangle \cdot \vec{S} \quad [A6]$$

where  $\vec{S}$  is the oriented surface of one interface and the mean value of the expression between brackets is taken over the interface. Since  $D(\vec{r})$  and  $\vec{\nabla}(C(\vec{r}, t)/v_e(\vec{r}))$  are not expected to vary significantly along the interface of two voxels, the mean value of their product can be approximated by the product of their extrapolated values at the interface.

$$\frac{d}{dt}\langle C(\vec{r}, t) \rangle V = \sum_{Neighbors} D(\vec{r}) \vec{\nabla} \left( \frac{C(\vec{r}, t)}{v_e(\vec{r})} \right) \Big|_{interface} \cdot \vec{S} \quad [A7]$$

Going from a continuous to a discrete formulation, Eq. [A7] becomes:

$$\frac{d}{dt} C_{i,j}(t) = \sum_{Neighbors} \left( \frac{D_{i,j} + D_{Neighbor}}{2} \right) \frac{1}{a} \left( \frac{C_{Neighbor}(t)}{v_{e Neighbor}} - \frac{C_{i,j}(t)}{v_{e i,j}} \right) \frac{S}{V} \quad [A8]$$

where  $a$  is the length of one side of a square prism voxel.

## REFERENCES

1. Kety SS. The theory and applications of the exchange of inert gas at the lungs and tissues. *Pharmacol Rev* 1951; 3:1–41.
2. Larsson HBW, Stubgaard M, Frederiksen JL, Jensen M, Henriksen O, Paulson OB. Quantitation of Blood-Brain Barrier Defect by Magnetic Resonance Imaging and Gadolinium-DTPA in Patients with Multiple Sclerosis and Brain Tumors. *Magn Reson Med* 1990; 16:117-131.
3. Brix G, Semmler W, Rüdiger P, Lothar RS, Günter L, Lorenz WJ. Pharmacokinetic Parameters in CNS Gd-DTPA Enhanced MR imaging. *J Comput Assist Tomogr* 1991; 15:621-628.
4. Tofts PS, Kermode AG. Measurements of the blood-brain barrier permeability and leakage space using dynamic MR imaging: 1.Fundamental concepts. *Magn Reson Med* 1991; 17:357-367.
5. Tofts PS. Modeling tracer kinetics in dynamic Gd-DTPA MR imaging. *J Magn Reson Imaging* 1997; 7:91-101.
6. Tofts PS, Brix G, Buckley DL, Evelhoch JL, Henderson E, Knopp MV, Larsson HBW, Lee T-Y, Mayr NA, Parker GJM, Port RE, Taylor J, Weisskoff RM. Estimating kinetic parameters from dynamic contrast-enhanced T1-weighted MRI of a diffusible tracer: standardized quantities and symbols. *J Magn Reson Imaging* 1999; 10:223-232.
7. Schlemmer H-P, Merkle J, Grobholz R, Jaeger T, Michel MS, Werner A, Rabe J, van Kaick G. Can pre-operative contrast-enhanced dynamic MR imaging for prostate cancer predict microvessel density in prostatectomy specimens? *Eur Radiol* 2004; 14:309-317.
8. Yankeelov TE, Rooney WD, Li X, Springer CS Jr.. Variation of the Relaxographic “Shutter-Speed” for Transcytolemmal Water Exchange Affects the CR Bolus-Tracking Curve Shape. *Magn Reson Med* 2003; 50:1151-1169.

9. Li X, Rooney WD, Springer CS Jr.. A Unified Magnetic Resonance Imaging Pharmacokinetic Theory: Intravascular and Extracellular Contrast Reagents. *Magn Reson Med* 2005; 54:1351-1359.
10. Kovar DA, Lewis M, Karczmar GS. A new method for imaging perfusion and contrast extraction fraction: input functions derived from reference tissues. *J Magn Reson Imaging* 1998; 8:1126-1134.
11. Yang C, Karczmar GS, Medved M, Stadler WM. Estimating the arterial input function using two reference tissues in dynamic contrast-enhanced MRI studies: Fundamental concepts and simulations. *Magn Reson Med* 2004; 52:1110-1117.
12. Yankeelov TE, Luci JJ, Lepage M, Li R, Debusk L, Lin PC, Price RR, Gore JC. Quantitative pharmacokinetic analysis of DCE-MRI data without an arterial input function: a reference region model. *Magn Reson Imaging* 2005; 23:519-529.
13. Yankeelov TE, DeBusk LM, Billheimer D, Luci JJ, Lin C, Price RR, Gore JC. Repeatability of a Reference Region Model for Analysis of Murine DCE-MRI Data at 7T. *J Magn Reson Imaging* 2006; 24:1140-1147.
14. Yankeelov TE, Cron GO, Addison C, Wallace JC, Wilkins RC, Pappas BA, Santyr GE, Gore JC. Comparison of a Reference Region Model to Direct Measurement of an AIF in the Analysis of DCE-MRI Data. *Magn Reson Med*, in press.
15. Jain RK. Transport of Molecules in the Tumor Interstitium: A Review. *Cancer Res* 1987; 47:3039-3051.
16. Jain RK. Determinants of Tumor Blood Flow: A Review. *Cancer Res* 1988; 48:2641-2658.
17. Eskey CJ, Koretsky AP, Domach MM, Jain RK.  $^2\text{H}$ -Nuclear Magnetic Resonance Imaging of Tumor Blood Flow: Spatial and Temporal Heterogeneity in a Tissue-isolated Mammary Adenocarcinoma. *Cancer Res* 1992; 52:6010-6019.

18. Su M-Y, Cheung Y-C, Fruehauf JP, Yu H, Nalcioğlu O, Mechetner E, Kyshtoobayeva A, Chen S-C, Hsueh S, McLaren CE, Wan Y-L. Correlation of dynamic contrast enhancement MRI parameters with microvessel density and VEGF for assessment of angiogenesis in breast cancer. *J Magn Reson Imaging* 2003; 18:467-477.
19. Cha S, Johnson G, Wadghiri YZ, Jin O, Babb J, Zagzag D, Turnbull DH. Dynamic contrast-enhanced perfusion MRI in mouse gliomas: Correlation with histopathology. *Magn Reson Med* 2003; 49:848-855.
20. Stevenson JP, Rosen M, Sun W, Gallagher M, Haller DG, Vaughn D, Giantonio B, Zimmer R, Petros WP, Stratford M, Chaplin D, Young SL, Schnall M, O'Dwer PJ. Phase I Trial of the Antivascular Agent Combretastatin A4 Phosphate on a 5-Day Schedule to Patients With Cancer: Magnetic Resonance Imaging Evidence for Altered Tumor Blood Flow. *J Clin Oncol* 2003; 21:4428-4438.
21. Evelhoch JL, LoRusso PM, He Z, DelProposto Z, Polin L, Corbett TH, Langmuir P, Wheeler C, Stone A, Leadbetter J, Ryan AJ, Blakey DC, Waterton JC. Magnetic Resonance Imaging Measurements of the Response of Murine and Human Tumors to the Vascular-Targeting Agent ZD6126. *Clin Can Res* 2004; 10:3650-3657.
22. Jain RK. Normalizing tumour vasculature with anti-angiogenic therapy: A new paradigm for combination therapy. *Nat Med* 2001; 7:987-989.
23. Vaupel P, Mayer A, Briest S, Höckel M. Oxygenation gain factor : A novel parameter characterizing the association between haemoglobin level and oxygenation status of breast cancers. *Cancer Res* 2003; 63:7634-7637.
24. Hunter A, Hendrikse A, Renan M, Abratt R. Does the tumor microenvironment influence radiation-induced apoptosis? *Apoptosis* 2006; 11:1727-1735.
25. Aarts E, Korts J. Simulated Annealing and Boltzman Machines; A Stochastic Approach to Combinatorial Optimization and Neural Computing. New-York: Wiley; 1989. 272 pages.

26. Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E. Equation of state calculation by fast computing machines. *J Chem Phys* 1953; 21:1087-1092
27. Donahue KM, Weisskoff RM, Parmelee DJ, Callahan RJ, Wilkinson RA, Mandeville JB, Rosen BR. Dynamic Gd-DTPA Enhanced MRI Measurement of Tissue Cell Volume Fraction. *Magn Reson Med* 1995; 34:423-432.
28. Padhani AR, Hayes C, Landau S, Leach MO. Reproducibility of quantitative dynamic MRI of normal human tissues. *NMR Biomed* 2002; 15:143-153.
29. Walker-Samuel S, Leach MO, Collins DJ. Evaluation of response to treatment using DCE-MRI: the relationship between initial area under the gadolinium curve (IAUGC) and quantitative pharmacokinetic analysis. *Phys Med Biol* 2006; 51:3593-3602.
30. Weinmann HJ, Laniado M, Mützel W. Pharmacokinetics of Gd-DTPA/Dimeglumine after intravenous injection into healthy volunteers. *Physiol Chem Phys Med NMR* 1984; 16:167-172
31. Lanari C, Lüthy I, Lamb CA, Fabris V, Pagano E, Helguero LA, Sanjuan N, Merani S, Molinolo AA. Five Novel Hormone-responsive Cell Lines Derived from Murine Mammary Ductal Carcinomas: *In Vivo* and *in Vitro* Effects of Estrogens and Progestins. *Cancer Res* 2001; 61:293-302.
32. Port RE, Knopp MV, Brix G. Dynamic Contrast-Enhanced MRI Using Gd-DTPA: Interindividual Variability of the Arterial Input Function and Consequences for the Assessment of Kinetics in Tumors. *Magn Reson Med* 2001; 45:1030-1038.
33. Esserman L, Hylton N, George T, Weidner N. Contrast-Enhanced Magnetic Resonance Imaging to Assess Tumor Histopathology and Angiogenesis in Breast Carcinoma. *The Breast Journal* 1999; 5:13-21.
34. Walker-Samuel S, Leach MO, Collins DJ. Reference tissue quantification of DCE-MRI data without a contrast agent calibration. *Phys Med Biol* 2007; 52:589-601.

35. Lee JH, Springer CS. Effects of Equilibrium Exchange on Diffusion-Weighted NMR Signals: The Diffusigraphic "Shutter-Speed". *Magn Reson Med* 2003; 49:450-458.



## CONCLUSIONS

---

L'ensemble des travaux reporté dans ce mémoire visait à inclure de façon explicite la diffusion de l'agent de contraste dans la modélisation pharmacocinétique des données d'IRM-dynamique. Pour ce faire, un nouveau modèle combinant la perfusion et la diffusion a été développé et mis à l'épreuve. Les interactions entre les pixels découlant de la diffusion de l'agent ont forcé l'utilisation de méthodes d'optimisation aléatoire pour l'ajustement des données. Ainsi les algorithmes génétiques et le recuit simulé ont été employés pour l'ajustement de données simulées.

Les AG ont montré qu'ils pouvaient effectuer une recherche à travers le vaste espace des solutions. Les impacts de la taille de la population (PS) et de la probabilité de *crossover* (CP) ont été identifiés. La taille de la population joue un rôle important sur le temps d'exécution de l'algorithme. En effet, le nombre de générations complétées est linéairement proportionnel à la taille de la population. D'un autre côté, la diversité des caractéristiques des solutions est d'autant plus grande que le nombre d'individus dans la population est élevé. Pour sa part, la probabilité de *crossover* avait un rôle moins déterminant. Elle régula la vitesse de la descente initiale et elle avait une légère influence sur l'aptitude finale de la meilleure solution. Les AG

étant plus appropriés pour le calcul parallèle, ils ont rapidement été laissés de côté au profit du recuit simulé qui trouvait des solutions beaucoup plus adéquates pour le même temps de calcul.

Le recuit simulé a montré qu'il pouvait retrouver les valeurs réelles des deux paramètres,  $K^{\text{trans}}$  et  $v_e$ , utilisés pour la fabrication de données réalistes incluant la diffusion de l'AC dans les tissus. En effet, l'écart moyen des valeurs de  $K^{\text{trans}}$  était de 16% et 17 % pour  $v_e$ . Même en présence d'un niveau de bruit réaliste, l'algorithme parvenait à trouver des valeurs sensées pour les paramètres. Lorsque ces données simulées étaient ajustées avec un modèle standard à deux compartiments, les cartes de  $K^{\text{trans}}$  obtenues ne montraient pas la démarcation franche de la périphérie avec des valeurs plus élevées que le cœur. Dans ce cas, l'écart moyen des valeurs de  $K^{\text{trans}}$  atteignait 43%.

Lorsqu'appliqué à des données d'adénocarcinomes mammaires de souris, le modèle proposé a donné des cartes de  $K^{\text{trans}}$  avec une démarcation claire entre la périphérie et le cœur en accord avec le rehaussement initial perçu dans les images d'IRM. À l'opposé, le modèle standard à deux compartiments retournait encore une fois des cartes de  $K^{\text{trans}}$  qui paraissaient moyennées. De plus le modèle standard trouvait des valeurs de  $v_e$  sans sens physique ( $v_e > 1$ ) dans le cœur des tumeurs qui ne montrait pas de rehaussement initialement, mais qui accumulait l'AC par diffusion après un

certain temps. Le modèle proposé, pour sa part, retournait des valeurs sensées pour l'ensemble du tissu.

Les travaux présentés dans ce texte ont montré la faisabilité de la modélisation de la diffusion de l'AC présente dans les tissus. Dans le but de d'améliorer l'utilisation du modèle, il faudrait que l'algorithme ainsi que le code soient révisés pour en accélérer l'exécution. De plus, le coefficient de diffusion de l'agent de contraste a été posé comme étant égale à celui apparent de l'eau. Cette hypothèse devrait être développée pour trouver une meilleure correspondance entre ces deux quantités. Finalement, les AG pourraient être réutilisés dans le cadre du calcul parallèle pour tenter d'en améliorer les performances.

## REMERCIEMENTS

---

J'aimerais remercier Martin Lepage pour son support tout au long de mes travaux de maîtrise, pour son expertise et pour la précieuse complicité que j'ai pu développer avec lui.

Je remercie aussi Luc Tremblay pour de nombreuses discussions enrichissantes, son expertise en résonance magnétique et son amitié.

Il faut que je souligne la participation important de Mélanie Archambault, Céléna Dubuc, Véronique Dumulon et Simon Authier pour la préparation et la manipulation des sujets animaux.

Je remercie aussi les membres du département pour l'excellente ambiance de travail et les amitiés que j'y ai formées.

Je voudrais aussi remercier Michèle, mon amie de cœur, pour son support ainsi que mes parents, Richard et Renée, pour beaucoup plus de chose que je ne peux écrire ici.

Je dois souligner que pour entreprendre ma maîtrise, j'ai bénéficié d'une bourse d'études supérieures du CRSNG.

## **LISTE DES RÉFÉRENCES**

---

Aarts E et Korst J, Simulated Annealing and Boltzmann Machines : A Stochastic Approach to Combinatorial Optimization and Neural Computing, Wiley, 1989, 272 pages

Brix G, Semmler W, Rüdiger P, Lothar RS, Günter L and Lorenz WJ. Pharmacokinetic Parameters in CNS Gd-DTPA Enhanced MR imaging. J Comput Assist Tomogr 1991; 15:621-628.

Cha S, Johnson G, Wadghiri YZ, Jin O, Babb J, Zagzag D and Turnbull DH. Dynamic contrast-enhanced perfusion MRI in mouse gliomas: Correlation with histopathology. Magn Reson Med 2003; 49:848-855.

Damadian R, Tumor Detection by Nuclear Magnetic Resonance. Science 1971; 171: 1151-1153.

Donahue KM, Weisskoff RM, Parmelee DJ, Callahan RJ, Wilkinson RA, Mandeville JB and Rosen BR. Dynamic Gd-DTPA Enhanced MRI Measurement of Tissue Cell Volume Fraction. Magn Reson Med 1995; 34:423-432.

Eskey CJ, Koretsky AP, Domach MM and Jain RK.  $^2\text{H}$ -Nuclear Magnetic Resonance Imaging of Tumor Blood Flow: Spatial and Temporal Heterogeneity in a Tissue-isolated Mammary Adenocarcinoma. Cancer Res 1992; 52:6010-6019.

Fritz-Hansen T, Rostrup E, Larsson HBW, Sondergaard L, Ring P, Henrikson O. Measurement of the arterial concentration of Gd-DTPA using MRI: a step toward quantitative perfusion imaging. Magn Reson Med 1996; 36:225– 231.

Garroway AN, Grannell PK et Mansfield P. Image formation in NMR by selective irradiation process. J. Phys. C: Solid State Phys. 1974; 7:L457-L462.

Goldberg D. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley Longman, 1989, 412 pages

Haacke EM, Brown RW, Thompson MR et Venkatesan R. Magnetic Resonance Imaging : Physical Principles and Sequence Design. Wiley-Liss. New-York 1999, 914 pages.

Jain RK. Determinants of Tumor Blood Flow: A Review. Cancer Res 1988; 48:2641-2658.

Jain RK. Normalizing tumour vasculature with anti-angiogenic therapy: A new paradigm for combination therapy. Nat Med 2001; 7:987-989.

Jain RK. Transport of Molecules in the Tumor Interstitium: A Review. Cancer Res 1987; 47:3039-3051.

Lanari C, Lüthy I, Lamb CA, Fabris V, Pagano E, Helguero LA, Sanjuan N, Merani S and Molinolo AA. Five Novel Hormone-responsive Cell Lines Derived from Murine Mammary Ductal Carcinomas: *In Vivo* and *in Vitro* Effects of Estrogens and Progestins. Cancer Res 2001; 61:293-302.

Larsson HBW, Stubgaard M, Frederiksen JL, Jensen M, Henriksen O and Paulson OB. Quantitation of Blood-Brain Barrier Defect by Magnetic Resonance Imaging and Gadolinium-DTPA in Patients with Multiple Sclerosis and Brain Tumors. Magn Reson Med 1990, 16:117-131.

Lauterbur PC. Image Formation by Induced Local Interactions: Examples Employing Nuclear Magnetic Resonance. Nature 1973; 242:190-191.

Li X, Rooney WD and Springer CS Jr.. A Unified Magnetic Resonance Imaging Pharmacokinetic Theory: Intravascular and Extracellular Contrast Reagents. Magn Reson Med 2005; 54:1351-1359.

Metropolis N, Rosenbluth A, Rosenbluth M, Teller A and Telle E. Equation of state calculation by fast computing machines. J Chem Phys 1953; 21:1087-1092

Nishimura DG. Principles of Magnetic Resonance Imaging. 1996. 223 pages.

Padhani AR, Hayes C, Landau S and Leach MO. Reproducibility of quantitative dynamic MRI of normal human tissues. NMR Biomed 2002; 15:143-153.

Port RE, Knopp MV and Brix G. Dynamic Contrast-Enhanced MRI Using Gd-DTPA: Interindividual Variability of the Arterial Input Function and Consequences for the Assessment of Kinetics in Tumours, *Magn Reson Med* 2001; 45:1030-1038.

Roland M et Tozer TN. Clinical pharmacokinetics: concepts and applications, 2<sup>nd</sup> Edition, Philadelphia; Williams & Wilkins; 1994

Stanisz GL et Hankelman RM. Gd-DTPA Relaxivity Depends on Macromolecular Content. *Magn Reson Med* 2000; 44:665-667.

Su M-Y, Cheung Y-C, Fruehauf JP, Yu H, Nalcioğlu O, Mechetner E, Kyshtoobayeva A, Chen S-C, Hsueh S, McLaren CE and Wan Y-L. Correlation of dynamic contrast enhancement MRI parameters with microvessel density and VEGF for assessment of angiogenesis in breast cancer. *J Magn Reson Imaging* 2003; 18:467-477.

Tofts PS and Kermode AG. Measurements of the blood-brain barrier permeability and leakage space using dynamic MR imaging: 1. Fundamental concepts. *Magn Reson Med* 1991; 17:357-367.

Tofts PS, Brix G, Buckley DL, Evelhoch JL, Henderson E, Knopp MV, Larsson HBW, Lee T-Y, Mayr NA, Parker GJM, Port RE, Taylor J, and Weisskoff RM. Estimating kinetic parameters from dynamic contrast-enhanced T1-weighted MRI of a diffusible tracer: standardized quantities and symbols. *J Magn Reson Imaging* 1999; 10:223– 32.

Tofts PS. Modeling Tracer Kinetics in Dynamic Gd-DTPA MR Imaging. *J Magn Reson Imaging* 1997; 7: 91 – 101

Weinmann HJ, Laniado M and Mützel W. Pharmacokinetics of Gd-DTPA/Dimeglumine after intravenous injection into healthy volunteers. *Physiol Chem Phys Med NMR* 1984; 16:167-17

Yankeelov TE, Luci JJ, Lepage M, Li R, Debusk L, Lin PC, Price RR and Gore JC. Quantitative pharmacokinetic analysis of DCE-MRI data without an arterial input function: a reference region model, *Magn Reson Imaging* 2005; 23:519-529.

Yankeelov TE, Rooney WD, Li X and Springer CS Jr.. Variation of the Relaxographic "Shutter-Speed" for Transcytolemmal Water Exchange Affects the CR Bolus-Tracking Curve Shape. Magn Reson Med 2003; 50:1151-1169.



## ANNEXE A : IMAGES PONDÉRÉES PAR LA DIFFUSION

---

La démonstration ci-dessous est inspirée de HAACKE *et al.*, 1999.

En ajoutant la conservation de la masse à l'équation différentielle décrivant la magnétisation exposée à un champ magnétique on obtient :

$$\frac{\partial}{\partial t} \vec{M}(\vec{r}, t) = \gamma \vec{M}(\vec{r}, t) \times \vec{B} + D \nabla^2 \vec{M}(\vec{r}, t) \quad [A1]$$

Dans cette dernière équation,  $\vec{M}(\vec{r}, t) = M_x(\vec{r}, t) + iM_y(\vec{r}, t)$  et  $D$  est le coefficient de diffusion apparent de l'eau. La décroissance exponentielle en  $T_2$  ainsi que la précession engendrée par le champ  $B_0$  sont laissées de côté pour simplifier la démonstration. Si  $\vec{B}$  a la forme  $\vec{B} \equiv \vec{G} \cdot \vec{r} \hat{z}$ , alors l'Eq. [A1] devient :

$$\frac{\partial}{\partial t} \vec{M}(\vec{r}, t) = -i\gamma(\vec{r} \cdot \vec{G}) \vec{M}(\vec{r}, t) + D \nabla^2 \vec{M}(\vec{r}, t) \quad [A2]$$

La solution de cette équation sans le terme de diffusion est :

$$\vec{M}(\vec{r}, t) = A e^{-i\gamma \vec{r} \cdot \int_0^t \vec{G}(t') dt'} \quad [A3]$$

où  $A$  est une constante. Dans le cas où la diffusion ne dépend pas de la position, on aura que  $A \rightarrow A(t)$ . Ainsi

$$\frac{\partial}{\partial t} \vec{M}(\vec{r}, t) = e^{-i\gamma \vec{r} \cdot \int_0^t \vec{G}(t') dt'} \frac{\partial}{\partial t} A(t) - i\gamma \vec{r} \cdot \vec{G}(t) \vec{M}(\vec{r}, t) \quad [A4]$$

Avec l'équation [A2] on obtient donc :

$$\Rightarrow \frac{\partial}{\partial t} A(t) = e^{i\gamma \vec{r} \cdot \int_0^t \vec{G}(t') dt'} D \nabla^2 \vec{M}(\vec{r}, t) \quad [\text{A5}]$$

$$\Rightarrow \frac{1}{A(t)} \frac{\partial}{\partial t} A(t) = e^{i\gamma \vec{r} \cdot \int_0^t \vec{G}(t') dt'} D \nabla^2 e^{-i\gamma \vec{r} \cdot \int_0^t \vec{G}(t') dt'} \quad [\text{A6}]$$

$$\Rightarrow \frac{d}{dt} (\ln A(t)) = -D \left( \gamma \int_0^t \vec{G}(t') dt' \right)^2 \quad [\text{A7}]$$

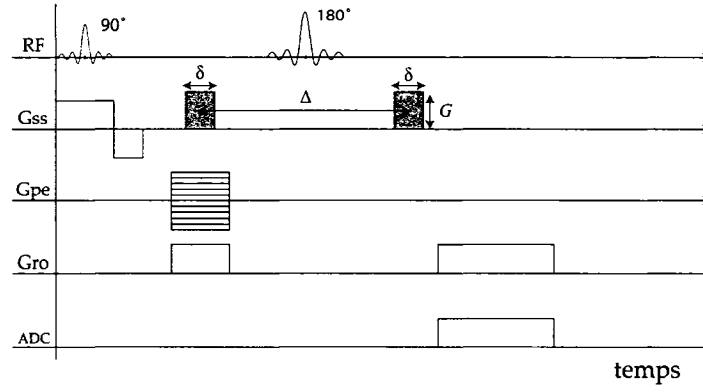
$$\Rightarrow \ln A(t) = -\gamma^2 D \int_0^t dt'' \left( \int_0^{t''} \vec{G}(t') dt' \right)^2 + \ln A(0) \quad [\text{A8}]$$

En posant  $A(0) = 1$ , le terme  $\ln A(0)$  peut être ignoré (Eq [A9]) sans changer les conclusions de cette démarche.

$$\Rightarrow \ln A(t) = -\gamma^2 D \int_0^t dt'' \left( \int_0^{t''} \vec{G}(t') dt' \right)^2 \quad [\text{A9}]$$

La séquence de pulse de la figure A1 est celle d'un écho de spin pondéré par la diffusion apparente de l'eau (Méthode Stejskal-Tanner). Dans cette séquence, la contribution des gradients de diffusion (en gris) se traduit par :

$$\int_0^{t''} \vec{G}(t') dt' = \begin{cases} GT'' & \text{pour } 0 \leq t'' \leq \delta \\ G\delta & \text{pour } \delta \leq t'' \leq T_E / 2 \\ -G\delta & \text{pour } T_E / 2 \leq t'' \leq \Delta \\ -G\delta + G(t'' - \Delta) & \text{pour } \Delta \leq t'' \leq \Delta + \delta \end{cases} \quad [\text{A10}]$$



**Figure A1** : Séquence de pulse d'un écho de spin pondéré par la diffusion apparente de l'eau. Ici  $G_{ss}$ ,  $G_{pe}$  et  $G_{ro}$  sont associés aux trois gradients orthogonaux utilisés en IRM (SS : Slice Select, PE : Phase Encoding et RO : Readout). L'ADC décrit la période d'acquisition du convertisseur analogique-digital (Analogue Digital Converter) de l'appareil d'IRM.

Ainsi

$$\ln A(t) = -\gamma^2 D \left[ \int_0^{\delta} G^2 t'^2 dt'' + \int_{\delta}^{T_E/2} G^2 \delta^2 dt'' + \int_{T_E/2}^{\Delta} (-G)^2 \delta^2 dt'' + \int_{\Delta}^{\Delta+\delta} G^2 (t'' - (\Delta + \delta))^2 dt'' \right] \quad [A11]$$

$$\ln A(t) = -\gamma^2 D G^2 \left[ \frac{\delta^3}{3} + \delta^2 \left( \frac{T_E}{2} - \delta \right) + \delta^2 \left( \Delta - \frac{T_E}{2} \right) + \frac{(\Delta + \delta)^3}{3} - (\Delta + \delta)^3 + (\Delta + \delta)^3 - \left( \frac{\Delta^3}{3} - \Delta^2 (\Delta + \delta) + \Delta (\Delta + \delta)^2 \right) \right] \quad [A12]$$

$$\ln A(t) = -\gamma^2 D G^2 \delta^2 \left( \Delta - \frac{\delta}{3} \right) \quad [A13]$$

La solution à l'Eq. [A1] est donc :

$$\bar{M}(\vec{r}, t) = A_0 e^{-\gamma^2 D G^2 \delta^2 \left( \Delta - \frac{\delta}{3} \right)} e^{-i\gamma \vec{r} \cdot \int_0^t \vec{G}(t') dt'} = A_0 e^{-D b} e^{-i\gamma \vec{r} \cdot \int_0^t \vec{G}(t') dt'} \quad [\text{A14}]$$

où  $b$  est le  $b$ -value [ $\text{mm}^{-2} \text{s}$ ]. L'Eq. [A13] peut être utilisée pour ajuster la valeur de l'intensité du signal pour plusieurs images avec des valeurs de  $b$  différentes. Cette approche permet de connaître le coefficient apparent de l'eau dans les tissus imagés. Pour plus de détails, voir HAACKE *et al.*, 1999.

## ANNEXE B : DÉFINITIONS DE MATRICE

$$\bar{K} = \begin{pmatrix} k_{11} & 0 & \dots & 0 \\ 0 & k_{12} & \dots & \dots \\ \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & k_{mn} \end{pmatrix}$$

$$\bar{V} = \begin{pmatrix} 1/v_{11} & 0 & \dots & 0 \\ 0 & 1/v_{12} & \dots & \dots \\ \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & 1/v_{mn} \end{pmatrix}$$

Eq B1

$$\bar{D} = \begin{pmatrix} -\frac{D_{2,1}+D_{1,2}+2D_{1,1}}{2} & \frac{D_{1,1}+D_{1,2}}{2} & 0 & \dots & 0 & \frac{D_{1,1}+D_{2,1}}{2} & 0 & \dots & \dots & 0 \\ \frac{D_{1,2}+D_{1,1}}{2} & -\frac{D_{2,2}+D_{1,3}+D_{1,1}+3D_{1,2}}{2} & \frac{D_{1,2}+D_{1,3}}{2} & 0 & \dots & 0 & \frac{D_{1,2}+D_{2,2}}{2} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \frac{D_{m,n-1}+D_{m-1,n-1}}{2} & 0 & \dots & 0 & \frac{D_{m,n-1}+D_{m,n-2}}{2} & -\frac{D_{m-1,n-1}+D_{m,n-2}+D_{m,n}+3D_{m,n-1}}{2} & \frac{D_{m,n-1}+D_{m,n}}{2} \\ 0 & \dots & \dots & 0 & \frac{D_{m,n}+D_{m-1,n}}{2} & 0 & \dots & 0 & \frac{D_{m,n}+D_{m,n-1}}{2} & -\frac{D_{m-1,n}+D_{m,n-1}+2D_{m,n}}{2} \end{pmatrix}$$

## ANNEXE C : CODE POUR LES ALGORITHMES GÉNÉTIQUES

```
function GenAlgo(seed,NI,PS,SE,CP,MP)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création des donnees de concentration
Data = genererDonnees(); % Data contient 13 mesures de la concentration, 121 mesures de Cp, la carte de diffusion et les variables x et y.

algorithmeGenetique(Data,[],str2num(seed),str2num(NI),str2num(PS),str2num(SE),str2num(CP),str2num(MP)); %algorithmeGenetique(Data,Population,seed pour le générateur de nombre aléatoire);
% NI : nombre d'itération, PS : PopulationSize, SE : SizeElistisme, CP : CrossoverProbability, MP : MutationProbability.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Data = genererDonnees
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initialisation

load parameters;

for ii = 1:size(k,1)
    for jj = 1:size(k,2)
        x(ii,jj) = ii;
        y(ii,jj) = jj;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Transformation de la concentration en vecteur et des parametres en
% matrices creuses.

counter = 1;
for ii = 1:size(x,1)
    for jj = 1:size(y,2)
        sparseK.aa(counter) = counter;    ,sparseK.bb(counter) = counter;
        ,sparseK.value(counter) = k(ii,jj);
        sparseV.aa(counter) = counter;    ,sparseV.bb(counter) = counter;
        ,sparseV.value(counter) = 1 / v(ii,jj);

        C(counter,1) = 0; % Passage de matrice m par n à vecteur de dimension mXn par 1
        counter = counter + 1;
    end
end
```

```

K = sparse(sparseK.aa,sparseK.bb,sparseK.value,size(x,1)*size(y,2),size(x,1)*size(y,2)); % Passage de matrice m par n à matrice creuse de dimension mXn par mXn
V = sparse(sparseV.aa,sparseV.bb,sparseV.value,size(x,1)*size(y,2),size(x,1)*size(y,2)); % Passage de matrice m par n à matrice creuse de dimension mXn par mXn

clear sparseK sparseV counter ii jj

% Création de la matrice creuse décrivant la diffusion

D = SparsingD(d);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Lecture de la concentration sanguine. Cp est un vecteur colonne avec 121 valeurs.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load Cp

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création du vecteur temps.

t = 0.1:0.1:length(Cp)/10;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création des concentrations
one(1:size(x,1) * size(y,2),1) = 1;

A = ( eye(size(x,1)*size(y,2)) - 0.1*K*V + 0.1*D*V );

B = 0.1*K*one;

for nn = 2:length(t)

    C(:,nn) = A * C(:,nn-1) + B*Cp(nn-1);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Préparation des données a retourner
for nn = 1:13
    Data.C(:,nn) = C(:,(nn-1)*10+1); % La fonction renvoie une mesure de C sur 10
end

```

```

Data.Cp = Cp; % La fonction renvoie la fonction d'entrée artérielle.
Data.d = d; % la fonction renvoie la carte de coefficients de de
diffusion
Data.x = x; % La fonction renvoie aussi la grille de x et de y pour que
le suivi des dimensions soit plus facile.
Data.y = y;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function BestIndividuals = algorithmeGenetique(Data,Pop,seed,NI,PS,SE,CP,
MP)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initialisation du générateur de nombre aléatoire
rand('state',seed);
clear seed;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création de la grille
[Grille.X Grille.Y] = find(~Data.d==0);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Parametres de l'otimisation par algorithmes génétiques
NombreIterations = NI;
PopSize = PS;
SizeElitism = round(SE*PopSize);
CrossoverProbability = CP;
MutationProbability = MP;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création de la population initiale si la fonction a été appelée sans
% population comme argument
if isempty(Pop) % Creation d'une population aléatoire si le fonction a
été appelée sans popuation comme argument.
    for nn = 1:PopSize
        % Attribution des valeurs de K
        Pop(nn).K(:, :) = GeneCreator(length(Grille.X),6,10,5,1,[]); %
Gene = GeneCreator(NumGene, GeneLength, Moy, Sigma, Density, min)

        % Attribution des valeurs de V
        Pop(nn).V(:, :) = GeneCreator(length(Grille.X),7,26,5,1,5); % Gene
= GeneCreator(NumGene, GeneLength, Moy, Sigma, Density, min)

    end

    clear nn

```



```

else
    PopSize = size(Pop,2); % Réajustement de la taille de la population ✓
    en fonction de celle passée en argument.
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création de la distribution de chance de reproduction. Ce vecteur
% déterminera la forme que prendra la pression de l'environnement sur la
% reproduction des individus.

% Dans le cas codé ci-dessous, les chances de reproduction d'un individu
% sont associées à son rang dans la population. Cette méthode prévient
% l'apparition de super individu qui domineraient la population après
% quelques génération, mais elle favorise aussi des individus médiocres
% dans une population médiocre.

rang(:,1) = 1:PopSize;
Chance(1,1) = 1;
for ii=2:PopSize
    Chance(ii,1) = 1 - 2 * sum(rang(PopSize-ii+2:PopSize)) / (PopSize*✓
(PopSize+1));
end
clear rang ii

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création de la matrice creuse décrivant la diffusion
d = Data.d;

D = SparsingD(d);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Évaluation de l'aptitude des individus de la population initiale.

for nn = 1:PopSize
    E(nn,1) = Aptitude(Pop(nn),Data,D,Grille);
end

clear nn
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Lancement de l'optimisation par algorithme génétique
tic % Pour Garder un suivi de la progression

for aa = 1 : NombreIterations

```

```

% Classement des individus
[SortedE,rang] = sort(E);

% L'aptitude des futurs individus est préalablement fixée à une
valeur
% arbitrairement élevée.
Emoy = mean(E);
NewE(1:PopSize,1) = 10*Emoy;

% Survie des individus les plus apte
for nn = 1:SizeElitism
    NewPop(nn) = Pop(rang(nn));
    NewE(nn) = E(rang(nn));
end

clear nn

% Création des nouveaux individus. Chaque itération créera 2 nouveaux
% individus à partir du bagage génétique des 2 parents.
for nn = SizeElitism + 1 : 2 :PopSize % Saut de 2 à cause qu'il y a 2
enfants

    while(NewE(nn,1) > 5 * Emoy | isnan(NewE(nn,1)) | NewE(nn+1,1) >
5 * Emoy | isnan(NewE(nn+1,1))) % Pour éviter les "freaks of nature" qui
nuiraient aux calculs

        % Sélection des parents
        Parent1 = rang( find(Chance >= rand,1,'last') );
        Parent2 = rang( find(Chance >= rand,1,'last') );

        % Initialisation des deux descendants
        Child1 = Pop(Parent1);
        Child2 = Pop(Parent2);

        % Mélange des genes de certains chromosomes;

        % Crée deux matrices avec les chromosomes à coupler dans la
% première colone et les sites de coupe dans le chromosome
dans
        % la deuxième colone.
        ToCrossK(:,1) = sort( ceil( rand(1,round(CrossoverProbability
* size(Pop(1).K,1)) * size(Pop(1).K,1) ));
        ToCrossK(:,2) = 2 + floor( rand(1,round(CrossoverProbability
* size(Pop(1).K,1)) * (size(Pop(1).K,2)-1) ));

        ToCrossV(:,1) = sort( ceil( rand(1,round(CrossoverProbability
* size(Pop(1).V,1)) * size(Pop(1).V,1) ));
        ToCrossV(:,2) = 2 + floor( rand(1,round(CrossoverProbability
* size(Pop(1).V,1)) * (size(Pop(1).V,2)-1) ));

        % Applications des couplages génétiques dans les gènes K.

```

```

        for ii = 1:size(ToCrossK,1)
            % display([Child1.K(ToCrossK(ii,1),:)' ' Child2.K
(ToCrossK(ii,1),:)' ' num2str(ToCrossK(ii,2)) ]);
            Child1.K(ToCrossK(ii,1),ToCrossK(ii,2):size(Child1.K,2))
= Pop(Parent2).K(ToCrossK(ii,1),ToCrossK(ii,2):size(Child1.K,2));
            Child2.K(ToCrossK(ii,1),ToCrossK(ii,2):size(Child1.K,2))
= Pop(Parent1).K(ToCrossK(ii,1),ToCrossK(ii,2):size(Child1.K,2));
            % display([Child1.K(ToCrossK(ii,1),:)' ' Child2.K
(ToCrossK(ii,1),:)]);
            % display(' ')
        end % for ii

% Applications des couplages génétiques dans les gènes V.
for ii = 1:size(ToCrossV,1)
    Child1.V(ToCrossV(ii,1),ToCrossV(ii,2):size(Child1.V,2))
= Pop(Parent2).V(ToCrossV(ii,1),ToCrossV(ii,2):size(Child1.V,2));
    Child2.V(ToCrossV(ii,1),ToCrossV(ii,2):size(Child1.V,2))
= Pop(Parent1).V(ToCrossV(ii,1),ToCrossV(ii,2):size(Child1.V,2));
    if bin2dec(Child1.V(ToCrossV(ii,1),:)) == 0 % Pour
prevenir les valeurs de V = 0 qui poseront des problèmes de division par
0.
        Child1.V(ToCrossV(ii,1),:) = Pop(Parent1).V(ToCrossV
(ii,1),:);
    end
    if bin2dec(Child2.V(ToCrossV(ii,1),:)) == 0 % Pour
prevenir les valeurs de V = 0 qui poseront des problèmes de division par
0.
        Child2.V(ToCrossV(ii,1),:) = Pop(Parent2).V(ToCrossV
(ii,1),:);
    end
end % for ii

clear ii ToCrossV ToCrossK

% Application des mutations

% Création de 4 matrices qui donneront les genes à modifier
% dans chaque génome.
ToMutateKC1 = rand(1 + round( MutationProbability*size
(Child1.K,1)*size(Child1.K,2) ),2);
ToMutateKC2 = rand(1 + round( MutationProbability*size
(Child2.K,1)*size(Child2.K,2) ),2);
ToMutateVC1 = rand(1 + round( MutationProbability*size
(Child1.V,1)*size(Child1.V,2) ),2);
ToMutateVC2 = rand(1 + round( MutationProbability*size
(Child2.V,1)*size(Child2.V,2) ),2);

for ii =1 : size(ToMutateKC1,1)
    if isequal(Child1.K(1+floor(ToMutateKC1(ii,1)*size
(Child1.K,1)),1+floor(ToMutateKC1(ii,2)*size(Child1.K,2))), '1')
        Child1.K(1+floor(ToMutateKC1(ii,1)*size(Child1.K,1)),

```

```

1+floor(ToMutateKC1(ii,2)*size(Child1.K,2))) = '0';
    else
        Child1.K(1+floor(ToMutateKC1(ii,1)*size(Child1.K,1)), 1+floor(ToMutateKC1(ii,2)*size(Child1.K,2))) = '1';
    end % if
end % for ii

for ii = 1 : size(ToMutateKC2,1)
    if isequal(Child2.K(1+floor(ToMutateKC2(ii,1)*size(Child2.K,1)), 1+floor(ToMutateKC2(ii,2)*size(Child2.K,2))), '1')
        Child2.K(1+floor(ToMutateKC2(ii,1)*size(Child2.K,1)), 1+floor(ToMutateKC2(ii,2)*size(Child2.K,2))) = '0';
    else
        Child2.K(1+floor(ToMutateKC2(ii,1)*size(Child2.K,1)), 1+floor(ToMutateKC2(ii,2)*size(Child2.K,2))) = '1';
    end % if
end % for ii

for ii = 1 : size(ToMutateVC1,1)
    if isequal(Child1.V(1+floor(ToMutateVC1(ii,1)*size(Child1.V,1)), 1+floor(ToMutateVC1(ii,2)*size(Child1.V,2))), '1')
        Child1.V(1+floor(ToMutateVC1(ii,1)*size(Child1.V,1)), 1+floor(ToMutateVC1(ii,2)*size(Child1.V,2))) = '0';
    else
        Child1.V(1+floor(ToMutateVC1(ii,1)*size(Child1.V,1)), 1+floor(ToMutateVC1(ii,2)*size(Child1.V,2))) = '1';
    end % if
    if bin2dec(Child1.V(1+floor(ToMutateVC1(ii,1)*size(Child1.V,1)), :)) == 0 % Pour prevenir les valeurs de V = 0 qui poseront des problèmes de division par 0.
        Child1.V(1+floor(ToMutateVC1(ii,1)*size(Child1.V,1)), :)) = Pop(Parent1).V(1+floor(ToMutateVC1(ii,1)*size(Child1.V,1)), :);
    end
end % for ii

for ii = 1 : size(ToMutateVC2,1)
    if isequal(Child2.V(1+floor(ToMutateVC2(ii,1)*size(Child2.V,1)), 1+floor(ToMutateVC2(ii,2)*size(Child2.V,2))), '1')
        Child2.V(1+floor(ToMutateVC2(ii,1)*size(Child2.V,1)), 1+floor(ToMutateVC2(ii,2)*size(Child2.V,2))) = '0';
    else
        Child2.V(1+floor(ToMutateVC2(ii,1)*size(Child2.V,1)), 1+floor(ToMutateVC2(ii,2)*size(Child2.V,2))) = '1';
    end % if
    if bin2dec(Child2.V(1+floor(ToMutateVC2(ii,1)*size(Child2.V,1)), :)) == 0 % Pour prevenir les valeurs de V = 0 qui poseront des problèmes de division par 0.
        Child2.V(1+floor(ToMutateVC2(ii,1)*size(Child2.V,1)), :)) = Pop(Parent2).V(1+floor(ToMutateVC2(ii,1)*size(Child2.V,1)), :);
    end
end % for ii

```

```

clear Parent1 Parent2 ToMutateKC1 ToMutateKC2 ToMutateVC1
ToMutateVC2 ii

% Évaluation de l'aptitude du nouvel individu
NewPop(nn) = Child1;
NewPop(nn+1) = Child2;

clear Child1 Child2

NewE(nn) = Aptitude(NewPop(nn),Data,D,Grille);
NewE(nn+1) = Aptitude(NewPop(nn+1),Data,D,Grille);

end % While

end % for nn

E = NewE;
Pop = NewPop;

Time = toc;

fid = fopen('GA_Status.dat','a');
fprintf(fid,'%6.6d\t %6.6d\t %4.0f\t %4.0f\n',mean(E),std(E),aa,
Time);
fclose(fid);

%La fonction sauvegarde la population finale
save Population Pop;

%La fonction sauvegarde le meilleur individu
Best = find(E==min(E),1);
BestSave(Pop(Best),E(Best),Grille,Data);

end % for aa

clear aa

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Ecart = Aptitude(Pop,Data,D,Grille)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Décodage de l'information contenu dans les chromosomes de l'individu.

Kchromosomes = bin2dec(Pop.K) / 100; % K est codé sur 6 bits de 0 à 64;
Vchromosomes = 1 ./ bin2dec(Pop.V) * 128; % V est codé sur 7 bits de 0 à

```

```

128;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

K = sparse(Grille.Y + (Grille.X-1) * size(Data.y,2),Grille.Y + (Grille.X-
1) * size(Data.y,2),Kchromosomes,size(Data.x,1)*size(Data.y,2),size(Data.
x,1)*size(Data.y,2)); % Passage d'un vecteur mxn par 1 à matrice creuse
de dimension mXn par mXn
V = sparse(Grille.Y + (Grille.X-1) * size(Data.y,2),Grille.Y + (Grille.X-
1) * size(Data.y,2),Vchromosomes,size(Data.x,1)*size(Data.y,2),size(Data.
x,1)*size(Data.y,2)); % Passage d'un vecteur mxn par 1 à matrice creuse
de dimension mXn par mXn

Cp(:,1) = Data.Cp;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création du vecteur temps.

t = 0.1:0.1:length(Cp)/10;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Creation des concentrations
one(1:size(Data.x,1) * size(Data.y,2),1) = 1;

%C(1:size(Data.x,1) * size(Data.y,2),1) = 0;
C1 = 1:size(Data.x,1) * size(Data.y,2);
C2(1:size(Data.x,1) * size(Data.y,2)) = 1;
Cvalue(1:size(Data.x,1) * size(Data.y,2)) = 0;

C = sparse(C1,C2,Cvalue);

A = sparse(( eye(size(Data.x,1)*size(Data.y,2)) - 0.1*K*V + 0.1*D*V ));

B = sparse(0.1*K*one);

for nn = 2:length(t)
    C(:,nn) = A * C(:,nn-1) + B*Cp(nn-1);
end

clear nn one K V D Cp Pop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calcul de l'aptitude (Ecart) de l'individu

for nn = 1:13
    Model(:,nn) = C(:,(nn-1)*10+1); % Les données initiales n'ont qu'une
mesure sur 10.
end

```

```

Ecart = 3 * sum((Data.C(:,3) - Model(:,3)).^2) / sum(Data.C(:,3)) + sum
((Data.C(:,13) - Model(:,13)).^2)/sum(Data.C(:,13)); %sum(sum((Data.C -
Model).^2));

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

function Gene = GeneCreator(NumGene, GeneLength, Moy, Sigma, Density, min)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

A(:,1) = round (abs(Moy + Sigma * randn(NumGene,1) ));

```

```

ToNull(1,:) = ceil(NumGene * rand(1, round(NumGene*(1-Density)) ));

```

```

for ii = 1:length(ToNull)
    A(ToNull(ii)) = 0;
end

```

```

if ~isempty(min)
    x = find(A < min);

```

```

    for ii = 1:length(x)
        A(x(ii)) = min;
    end
end

```

```

Gene = dec2bin(A, GeneLength);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

function SparseD = SparsingD(d)

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

counter = 1;
diag = 0;
for ii = 1:size(d,1)
    for jj = 1:size(d,2)
        if ~d(ii,jj)==0
            if ii - 1 > 0 & ~d(ii-1,jj)==0
                sparseD.aa(counter) = (ii-1)*size(d,2) + jj; ,sparseD.bb
(counter) = (ii-2)*size(d,2) + jj; ,sparseD.value(counter) = ( d(ii,jj) +
d(ii-1,jj) ) / 2;
                diag = diag - ( d(ii,jj) + d(ii-1,jj) ) / 2;
                counter = counter + 1;
            end
            if jj - 1 > 0 & ~d(ii,jj-1)==0
                sparseD.aa(counter) = (ii-1)*size(d,2) + jj; ,sparseD.bb
(counter) = (ii-1)*size(d,2) + jj-1; ,sparseD.value(counter) = ( d(ii,jj)

```

```

+ d(ii,jj-1) ) / 2;
    diag = diag - ( d(ii,jj) + d(ii,jj-1) ) / 2;
    counter = counter + 1;
end
if ii + 1 < size(d,1) + 1 & ~d(ii+1,jj)==0
    sparseD.aa(counter) = (ii-1)*size(d,2) + jj; ,sparseD.bb
(counter) = (ii)*size(d,2) + jj; ,sparseD.value(counter) = ( d(ii,jj) + d
(ii+1,jj) ) / 2;
    diag = diag - ( d(ii,jj) + d(ii+1,jj) ) / 2;
    counter = counter + 1;
end
if jj + 1 < size(d,2) + 1 & ~d(ii,jj+1)==0
    sparseD.aa(counter) = (ii-1)*size(d,2) + jj; ,sparseD.bb
(counter) = (ii-1)*size(d,2) + jj+1; ,sparseD.value(counter) = ( d(ii,jj)
+ d(ii,jj+1) ) / 2;
    diag = diag - ( d(ii,jj) + d(ii,jj+1) ) / 2;
    counter = counter + 1;
end
sparseD.aa(counter) = (ii-1)*size(d,2) + jj; ,sparseD.bb
(counter) = (ii-1)*size(d,2) + jj; ,sparseD.value(counter) = diag;
diag = 0;
counter = counter + 1;
end % ~d(ii,jj)==0
end
end

```

```

SparseD = sparse(sparseD.aa,sparseD.bb,sparseD.value,size(d,1)*size(d,2),
size(d,1)*size(d,2)); % Passage de matrice m par n à matrice creuse de
dimension mXn par mXn
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

function BestSave(Solution,BestE,Grille,Data)

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création des matrices de k et v
Best.k(1:size(Data.x,1),1:size(Data.y,2)) = 0;
Best.v(1:size(Data.x,1),1:size(Data.y,2)) = 0;

% Attribution des valeurs de k et v
counter = 1;
for ii=1:length(Grille.X)
    Best.k(Grille.X(ii),Grille.Y(ii)) = bin2dec(Solution.K(counter,:)) /
100;
    Best.v(Grille.X(ii),Grille.Y(ii)) = bin2dec(Solution.V(counter,:)) /
128;
    counter = counter +1 ;
end

```



```
% Sauvegarde de la meilleure solution
save BestSolution Best BestE Data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## ANNEXE D : CODE POUR LE RECUIT SIMULÉ

```
function SAnoise(seed,T0,DP,noiseLevel)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Création des donnees de concentration
Data = genererDonnees(); % Data contient 13 mesures de la concentration,
121 mesures de Cp, la carte de diffusion et les variables x et y.

% Ajout du bruit aux données
Data = NoisyData(Data,str2num(noiseLevel));

% Lancement de l'algorithme de recuit simulé
SimulatedAnnealing(Data,str2num(T0),str2num(DP),str2num(seed)); %
SimulatedAnnealing(Data,seed pour le générateur de nombres aléatoires,
température initiale,paramètre de descente)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

function Data = genererDonnees
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Lecture des valeurs originales de Ktrans, ve est ADC.
load parameters;

% Création de la trame des images
for ii = 1:size(k,1)
    for jj = 1:size(k,2)
        x(ii,jj) = ii;
        y(ii,jj) = jj;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Transformation de la concentration en vecteur et des parametres en
% matrices creuses.

counter = 1;
for ii = 1:size(x,1)
    for jj = 1:size(y,2)
        sparseK.aa(counter) = counter; ,sparseK.bb(counter) = counter;
        ,sparseK.value(counter) = k(ii,jj);
```

```

        sparseV.aa(counter) = counter;    ,sparseV.bb(counter) = counter;
,sparseV.value(counter) = 1 / v(ii,jj);

        C(counter,1) = 0; % Passage de matrice m par n à vecteur de
dimension mXn par 1
        counter = counter + 1;
    end
end

K = sparse(sparseK.aa,sparseK.bb,sparseK.value,size(x,1)*size(y,2),size
(x,1)*size(y,2)); % Passage de matrice m par n à matrice creuse de
dimension mXn par mXn
V = sparse(sparseV.aa,sparseV.bb,sparseV.value,size(x,1)*size(y,2),size
(x,1)*size(y,2)); % Passage de matrice m par n à matrice creuse de
dimension mXn par mXn

clear sparseK sparseV counter ii jj

% Création de la matrice creuse décrivant la diffusion
D = SparsingD(d*60); %La multiplication par 60 fait passer les
coefficients de diffusion de mm2/sec à mm2/min.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

% Lecture de la concentration sanguine. Cp est un vecteur colonne
contenant
% 121 éléments.
load Cp;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

% Création du vecteur temps.
t = 0.1:0.1:length(Cp)/10;
deltat = 0.1; % L'entré de l'agent de contraste sera calculé à chaque 6
secondes

deltatprime = 0.1 / 6; % La diffusion sera calculé à chaque seconde

power = 6; % Le nombre de pas de diffusion entre 2 admission d'agent de
contraste.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

% Format des pixels

```

```

a = 0.25; % en mm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Création des concentrations

% Vecteur de format m*n par 1 rempli de 1
one(1:size(x,1) * size(y,2),1) = 1;

% Calcul de la matrice DD qui comprend 6 pas de diffusion avec
deltatprime
% = 0.0167 min = 1 sec.
DDprime = sparse( eye(size(x,1)*size(y,2)) + deltatprime/a^2*D*V );
DD = DDprime;
for ii = 2:power
    DD = DD * DDprime;
end

% LA matrice A et B servent au calcul matriciel qui détermine C(t)
A = sparse((DD - deltat*K*V));

B = sparse(deltat*K*one);

for nn = 2:length(t)
    C(:,nn) = A * C(:,nn-1) + B*Cp(nn-1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Préparation des données a retourner
for nn = 1:13
    Data.C(:,nn) = C(:,(nn-1)*10+1); % La fonction renvoie une mesure de
C sur 10 pour représenter des données réelles.
    % Replace les concentration dans une matrice m par n
    counter = 1;
    for ii = 1:size(x,1)
        for jj = 1:size(y,2)
            c(ii,jj,1,nn) = Data.C(counter,nn);
            counter = counter + 1;
        end % ii
    end % jj
    clear counter
end
Data.Cp = Cp; % La fonction renvoie la fonction d'entrée artérielle.
Data.d = d; % la fonction renvoie la carte de coefficients de diffusion
Data.x = x; % La fonction renvoie aussi la grille de x et de y pour que

```

```

le suivi des dimensions soit plus facile.
Data.y = y;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

function Data = NoisyData(Data,noiseLevel,seed)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%Initialisation du générateur aléatoire
rand('state',1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

Data.C = Data.C + max(max(Data.C)) * randn(size(Data.C,1),size(Data.C,2)) *
* noiseLevel;
Data.Cp = Data.Cp + max(Data.Cp) * randn(length(Data.Cp),1) *
noiseLevel;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

function SimulatedAnnealing(Data,T0,DP,seed);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%Initialisation du générateur aléatoire
rand('state',seed);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Création de la grille
[Grille.X Grille.Y] = find(~Data.d==0);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Création de la solution initiale
% Seules les pixels avec une valeur de d (l'ADC) auront un gène attribué.
% Attribution des valeurs de K
Solution.K(:, :) = GeneCreator(length(Grille.X),6,10,5,1,[]); % Gene =
GeneCreator(NumGene,GenLength,Moy,Sigma,Density,min)

% Attribution des valeurs de V
Solution.V(:, :) = GeneCreator(length(Grille.X),7,26,5,1,5); % Gene =
GeneCreator(NumGene,GenLength,Moy,Sigma,Density,min)

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Création de la matrice creuse décrivant la diffusion
D = SparsingD(Data.d*60); %La multiplication par 60 fait passer les
coefficients de diffusion de mm2/sec à mm2/min.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Calcul de l'aptitude (fonction à minimiser) de la première solution
E = Aptitude(Solution,Data,D,Grille);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Calcul de la valeur initiale du parametre de contrôle (température)
% Les lignes suivantes peuvent être mise en commentaire si la valeur
% initiale de température est déjà connue.
%
% m1 = 0; % Nombre de transition vers un état de plus basse énergie
% m2 = 0; % Nombre de transistion vers un état de plus haute énergie
% meanDeltaEplus = 0;
% IAR = 0.95; % Initial acceptance ratio
%
%
% for ii = 1:1000
%
%     % Création d'un nouvel individu
%     % Sélection d'un site de modification
%     ParameterToModify = ceil(rand*2);
%     if ParameterToModify == 1
%         GeneToModify = ceil( rand * size(Solution.K,1) );
%         SiteToModify = ceil( rand * size(Solution.K,2) );
%     else
%         GeneToModify = ceil( rand * size(Solution.V,1) );
%         SiteToModify = ceil( rand * size(Solution.V,2) );
%     end
%
%     % Initialisation du nouvel individu
%     NewSolution = Solution;
%
%     % Modification du site
%     if ParameterToModify == 1
%         if isequal(Solution.K(GeneToModify,SiteToModify),'0')

```

```

%         NewSolution.K(GeneToModify,SiteToModify) = '1';
%     else
%         NewSolution.K(GeneToModify,SiteToModify) = '0';
%     end
% else
%     if isequal(Solution.V(GeneToModify,SiteToModify),'0')
%         NewSolution.V(GeneToModify,SiteToModify) = '1';
%     else
%         NewSolution.V(GeneToModify,SiteToModify) = '0';
%     end
%     % Pour empêcher l'apparition d'un "freak of nature".
%     if bin2dec(NewSolution.V(GeneToModify,:)) < 5
%         NewSolution.V(GeneToModify,SiteToModify) = Solution.V
(GeneToModify,SiteToModify);
%     end
% end
%
% % Évaluation de l'aptitude du nouvel individu.
% tic
% NewE = Aptitude(NewSolution,Data,D,Grille);
% toc
% display(num2str(ii));
%
% if NewE > 10 % To prevent the freak of nature
%     NewSolution = Solution;
%     NewE = E;
% end
%
% if NewE - E > 0
%     m2 = m2 + 1;
%     meanDeltaEplus = meanDeltaEplus + (NewE - E)/m2;
% else
%     m1 = m1 + 1;
% end
%
% InitialTemperature(ii) = meanDeltaEplus / log( m2 / (m2* IAR - m1*
(1-IAR)) );
% fid = fopen('Temperature.dat','a');
% fprintf(fid,'%4.0f \t %6.6f \n',ii,InitialTemperature(ii));
% fclose(fid);
%
% end % ii
%
% Temperature = InitialTemperature(length(InitialTemperature)); % Valeur
initiale du parametre de contrôle
%
% clear C ii m1 m2 meanDeltaEplus NewSolution NewE GeneToModify
SiteToModify

% Si la valeur initiale de température est déjà connue, elle sera
attribuée

```

```

% ici.
Temperature = T0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Optimisation par Simulated Annealing

NumberOfIterations = 0.50 * (size(Solution.K,1) * size(Solution.K,2) +
size(Solution.V,1) * size(Solution.V,2));

counter = 1;
counter2 = 1;
Fitness = 1;
tic;

Best = Solution;
BestE = E;

while Fitness(length(Fitness)) > 2e-5 % Condition irréalisable pour que
l'algorithme s'arrête

    ii = 1;
    NumberOfTries = 0;
    while ii < NumberOfIterations

        NumberOfTries = NumberOfTries + 1;

        % Création d'un nouvel individu
        % Sélection d'un site de modification
        ParameterToModify = ceil(rand*2);
        if ParameterToModify == 1
            GeneToModify = ceil( rand * size(Solution.K,1) );
            SiteToModify = ceil( rand * size(Solution.K,2) );
        else
            GeneToModify = ceil( rand * size(Solution.V,1) );
            SiteToModify = ceil( rand * size(Solution.V,2) );
        end

        % Initialisation du nouvel individu
        NewSolution = Solution;

        % Modification du site
        if Fitness(length(Fitness)) > 1e-5

            if ParameterToModify == 1
                if isequal(Solution.K(GeneToModify,SiteToModify), '0')
                    NewSolution.K(GeneToModify,SiteToModify) = '1';

```



```

        else
            NewSolution.K(GeneToModify,SiteToModify) = '0';
        end
    else
        if isequal(Solution.V(GeneToModify,SiteToModify), '0')
            NewSolution.V(GeneToModify,SiteToModify) = '1';
        else
            NewSolution.V(GeneToModify,SiteToModify) = '0';
        end
        % Pour empêcher l'apparition d'un "freak of nature".
        if bin2dec(NewSolution.V(GeneToModify,:)) < 5
            NewSolution.V(GeneToModify,SiteToModify) = Solution.V
(GeneToModify,SiteToModify);
        end
    end

else
    if ParameterToModify == 1
        if isequal(bin2dec(NewSolution.K(GeneToModify)),2^size
(NewSolution.K,2))
            NewSolution.K(GeneToModify,:) = dec2bin( bin2dec
(NewSolution.K(GeneToModify,:)) - 1,size(Solution.K,2));
        elseif isequal(bin2dec(NewSolution.K(GeneToModify)),0)
            NewSolution.K(GeneToModify,:) = dec2bin( bin2dec
(NewSolution.K(GeneToModify,:)) + 1,size(Solution.K,2));
        elseif rand > 0.5
            NewSolution.K(GeneToModify,:) = dec2bin( bin2dec
(NewSolution.K(GeneToModify,:)) + 1,size(Solution.K,2));
        else
            NewSolution.K(GeneToModify,:) = dec2bin( bin2dec
(NewSolution.K(GeneToModify,:)) - 1,size(Solution.K,2));
        end
    else
        if isequal(bin2dec(NewSolution.V(GeneToModify)),2^size
(NewSolution.V,2))
            NewSolution.V(GeneToModify,:) = dec2bin( bin2dec
(NewSolution.V(GeneToModify,:)) - 1,size(Solution.V,2));
        elseif isequal(bin2dec(NewSolution.V(GeneToModify)),5) %
5 est le minimum accepté pour éviter les freaks of nature
            NewSolution.V(GeneToModify,:) = dec2bin( bin2dec
(NewSolution.V(GeneToModify,:)) + 1,size(Solution.V,2));
        elseif rand > 0.5
            NewSolution.V(GeneToModify,:) = dec2bin( bin2dec
(NewSolution.V(GeneToModify,:)) + 1,size(Solution.V,2));
        else
            NewSolution.V(GeneToModify,:) = dec2bin( bin2dec
(NewSolution.V(GeneToModify,:)) - 1,size(Solution.V,2));
        end
    end

end
end

```

```

end

clear GeneToModify SiteToModify ParameterToModify

% Évaluation de l'aptitude du nouvel individu.
NewE = Aptitude(NewSolution,Data,D,Grille);

if NewE > 10 % To prevent the freak of nature
    NewSolution = Solution;
    NewE = E;
end

% Détermination du passage à ce nouvel individu
if NewE - E < 0
    Solution = NewSolution;
    E = NewE;
    EE(ii) = NewE;
    ii = ii + 1;
    % Pour conserver en mémoire la meilleure solution jusqu'à
    % présent.
    if NewE < BestE;
        BestSave(NewSolution,NewE,Grille,Data);
    end
elseif rand <= exp(-(NewE - E)/Temperature) & ~isequal(NewE,E)
    Solution = NewSolution;
    E = NewE;
    EE(ii) = NewE;
    ii = ii + 1;
end

end % while ii < NumberOfIterations

Fitness(counter) = E;
counter = counter + 1;
Time = toc;

Temperature = Temperature * DP;

fid = fopen('SA_Status.dat','a');
fprintf(fid,'%6.6d\t %6.6d\t %6.6d\t %6.6d\t %4.0f\t %4.0f\n',mean(
(E)),std(E),Temperature,NumberOfTries/NumberOfIterations,counter,Time);
fclose(fid);

clear ii EE fid

end % while Fitness(counter-1) > 2e-5

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

```

```

function Ecart = Aptitude(Pop,Data,D,Grille)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Décodage de l'information contenu dans les chromosomes de l'individu.

Kchromosomes = bin2dec(Pop.K) / 100; % K est codé sur 6 bits de 0 à 64;
Vchromosomes = 1 ./ bin2dec(Pop.V) * 128; % V est codé sur 7 bits de 0 à 128;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

K = sparse(Grille.Y + (Grille.X-1) * size(Data.y,2),Grille.Y + (Grille.X-1) * size(Data.y,2),Kchromosomes,size(Data.x,1)*size(Data.y,2),size(Data.x,1)*size(Data.y,2)); % Passage d'un vecteur mxn par 1 à matrice creuse de dimension mXn par mXn
V = sparse(Grille.Y + (Grille.X-1) * size(Data.y,2),Grille.Y + (Grille.X-1) * size(Data.y,2),Vchromosomes,size(Data.x,1)*size(Data.y,2),size(Data.x,1)*size(Data.y,2)); % Passage d'un vecteur mxn par 1 à matrice creuse de dimension mXn par mXn

Cp(:,1) = Data.Cp;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Création du vecteur temps.
t = 0.1:0.1:length(Cp)/10;
deltat = 0.1; % L'entrée de l'agent de contraste sera calculé à chaque 6 secondes

deltatprime = 0.1 / 6; % La diffusion sera calculé à chaque seconde

power = 6; % Le nombre de pas de diffusion entre 2 admission d'agent de contraste.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Format des pixels
a = 0.25; % en mm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

% Creation des concentrations
one(1:size(Data.x,1) * size(Data.y,2),1) = 1;

%C(1:size(Data.x,1) * size(Data.y,2),1) = 0;

```

```

C1 = 1:size(Data.x,1) * size(Data.y,2);
C2(1:size(Data.x,1) * size(Data.y,2)) = 1;
Cvalue(1:size(Data.x,1) * size(Data.y,2)) = 0;

C = sparse(C1,C2,Cvalue);

% Calcul de la matrice DD qui comprend 6 pas de diffusion avec
deltatprime
% = 0.0167 min = 1 sec.
DDprime = sparse( eye(size(Data.x,1)*size(Data.y,2)) +
deltatprime/a^2*D*V );
DD = DDprime;
for ii = 2:power
    DD = DD * DDprime;
end

% La matrice A et B servent au calcul matricielle qui détermine C(t)
A = sparse((DD - deltat*K*V));

B = sparse(deltat*K*one);

for nn = 2:length(t)
    C(:,nn) = A * C(:,nn-1) + B*Cp(nn-1);
end

clear nn one K V D Cp Pop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

% Calcul de l'aptitude de l'individu

for nn = 1:13
    Model(:,nn) = C(:,(nn-1)*10+1); % Les données initiales n'ont qu'une
mesure sur 10.
end

Ecart = 2 * sum((Data.C(:,3) - Model(:,3)).^2) / sum(Data.C(:,3)) + 2 *
sum((Data.C(:,4) - Model(:,4)).^2) / sum(Data.C(:,4)) + sum((Data.C(:,12)
- Model(:,12)).^2)/sum(Data.C(:,12)) + sum((Data.C(:,13) - Model(:,13)).
^2)/sum(Data.C(:,13)); %sum(sum((Data.C - Model).^2));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

function Gene = GeneCreator(NumGene, GeneLength, Moy, Sigma, Density, min)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%

A(:,1) = round (abs(Moy + Sigma * randn(NumGene,1) ));

ToNull(1,:) = ceil(NumGene * rand(1, round(NumGene*(1-Density)) ));

for ii = 1:length(ToNull)
    A(ToNull(ii)) = 0;
end

if ~isempty(min)
    x = find(A < min);

    for ii = 1:length(x)
        A(x(ii)) = min;
    end
end

Gene = dec2bin(A, GeneLength);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

function SparseD = SparsingD(d)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

counter = 1;
diag = 0;
for ii = 1:size(d,1)
    for jj = 1:size(d,2)
        if ~d(ii,jj)==0
            if ii - 1 > 0 & ~d(ii-1,jj)==0
                sparseD.aa(counter) = (ii-1)*size(d,2) + jj; ,sparseD.bb
(counter) = (ii-2)*size(d,2) + jj; ,sparseD.value(counter) = ( d(ii,jj) +
d(ii-1,jj) ) / 2;
                diag = diag - ( d(ii,jj) + d(ii-1,jj) ) / 2;
                counter = counter + 1;
            end
            if jj - 1 > 0 & ~d(ii,jj-1)==0
                sparseD.aa(counter) = (ii-1)*size(d,2) + jj; ,sparseD.bb
(counter) = (ii-1)*size(d,2) + jj-1; ,sparseD.value(counter) = ( d(ii,jj)
+ d(ii,jj-1) ) / 2;
                diag = diag - ( d(ii,jj) + d(ii,jj-1) ) / 2;
                counter = counter + 1;
            end
            if ii + 1 < size(d,1) + 1 & ~d(ii+1,jj)==0
                sparseD.aa(counter) = (ii-1)*size(d,2) + jj; ,sparseD.bb
(counter) = (ii)*size(d,2) + jj; ,sparseD.value(counter) = ( d(ii,jj) + d
(ii+1,jj) ) / 2;
            end
        end
    end
end

```

```

        diag = diag - ( d(ii,jj) + d(ii+1,jj) ) / 2;
        counter = counter + 1;
    end
    if jj + 1 < size(d,2) + 1 & ~d(ii,jj+1)==0
        sparseD.aa(counter) = (ii-1)*size(d,2) + jj; ,sparseD.bb
(counter) = (ii-1)*size(d,2) + jj+1; ,sparseD.value(counter) = ( d(ii,jj)
+ d(ii,jj+1) ) / 2;
        diag = diag - ( d(ii,jj) + d(ii,jj+1) ) / 2;
        counter = counter + 1;
    end
    sparseD.aa(counter) = (ii-1)*size(d,2) + jj; ,sparseD.bb
(counter) = (ii-1)*size(d,2) + jj; ,sparseD.value(counter) = diag;
    diag = 0;
    counter = counter + 1;
end % ~d(ii,jj)==0
end
end

```

```

SparseD = sparse(sparseD.aa,sparseD.bb,sparseD.value,size(d,1)*size(d,2),
size(d,1)*size(d,2)); % Passage de matrice m par n à matrice creuse de
dimension mXn par mXn
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

```

```

function BestSave(Solution,BestE,Grille,Data)

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Création des matrices de k et v
Best.k(1:size(Data.x,1),1:size(Data.y,2)) = 0;
Best.v(1:size(Data.x,1),1:size(Data.y,2)) = 0;

% Attribution des valeurs de k et v
counter = 1;
for ii=1:length(Grille.X)
    Best.k(Grille.X(ii),Grille.Y(ii)) = bin2dec(Solution.K(counter,:)) /
100;
    Best.v(Grille.X(ii),Grille.Y(ii)) = bin2dec(Solution.V(counter,:)) /
128;
    counter = counter +1 ;
end

Best.K = Solution.K;
Best.V = Solution.V;

% Sauvegarde de la meilleure solution
save BestSolution Best BestE Data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```